

---

---

# BASI DI DATI E CONOSCENZA

---

---

RELAZIONE DELLA PROGETTAZIONE DI UNA BASE DI DATI  
RELAZIONALE PER UNA COMPETIZIONE DI CROSSFIT

ANNO ACCADEMICO, 2015-2016

REALIZZATO DA

ROBERTO PALLOTTA  
*Universita degli studi di Roma  
Tor Vergata.*

# Indice

<b>1</b>	<b>Descrizione del progetto</b>	<b>4</b>
<b>2</b>	<b>Analisi dei requisiti</b>	<b>6</b>
<b>3</b>	<b>Regole Aziendali</b>	<b>8</b>
3.1	Regole di Vincolo . . . . .	8
3.2	Regole di Derivazione . . . . .	8
<b>4</b>	<b>Schema Entity-Relationship</b>	<b>8</b>
<b>5</b>	<b>Normalizzazione schema Entity-Relationship</b>	<b>12</b>
5.1	Prima forma normale. . . . .	12
5.2	Seconda Forma Normale . . . . .	12
5.3	Terza Forma Normale . . . . .	12
<b>6</b>	<b>Schema Fisico</b>	<b>12</b>
<b>7</b>	<b>Creazione delle Entità e delle Relazioni.</b>	<b>14</b>
<b>8</b>	<b>Inserimento dei dati nel database</b>	<b>17</b>
<b>9</b>	<b>Query</b>	<b>26</b>
9.1	Query 1 . . . . .	26
9.2	Query 2 . . . . .	26
9.3	Query 3 . . . . .	26
9.4	Query 4 . . . . .	27
9.5	Query 5 . . . . .	27
9.6	Query 6 . . . . .	27
9.7	Query 7 . . . . .	28
9.8	Query 8 . . . . .	28
9.9	Query 9 . . . . .	28
9.10	Query 10 . . . . .	29
9.11	Query 11 . . . . .	29
9.12	Query 12 . . . . .	29
9.13	Query 13 . . . . .	30
9.14	Query 14 . . . . .	30
9.15	Query 15 . . . . .	30
9.16	Query 16 . . . . .	31
9.17	Query 17 . . . . .	31
9.18	Query 18 . . . . .	31

<b>10 Algebra Relazionale</b>	<b>32</b>
10.1 Query 1 in Algebra relazionale . . . . .	32
10.2 Query 2 in Algebra Relazionale . . . . .	32
<b>11 Calcolo Relazionale</b>	<b>33</b>
11.1 Query 1 in Calcolo Relazionale . . . . .	33
11.2 Query 2 in Calcolo Relazionale . . . . .	33
<b>12 Trigger</b>	<b>34</b>
12.1 Trigger 1 . . . . .	34
12.2 Trigger 2 . . . . .	34
12.3 Trigger 3 . . . . .	35
<b>13 Stored Procedure</b>	<b>35</b>
13.1 Procedura 1 . . . . .	35
13.2 Procedura 2 . . . . .	37
13.3 Procedura 3 . . . . .	38
13.4 Procedura 4 . . . . .	39
<b>14 Ottimizzazioni</b>	<b>41</b>
14.1 ottimizzazione query 4 . . . . .	41
14.1.1 Query non ottimizzata . . . . .	41
14.1.2 Query ottimizzata . . . . .	41
14.1.3 Explain query non ottimizzata . . . . .	41
14.1.4 Explain query ottimizzata . . . . .	42
14.2 ottimizzazione query 7 . . . . .	43
14.2.1 Query non ottimizzata . . . . .	43
14.2.2 Query ottimizzata . . . . .	44
14.2.3 Explain query non ottimizzata . . . . .	44
14.2.4 Explain query ottimizzata . . . . .	44
14.3 ottimizzazione query 11 . . . . .	45
14.3.1 Explain query senza l'utilizzo dell'indice . . . . .	45
14.3.2 Explain query con l'utilizzo dell'indice . . . . .	46

## Elenco delle figure

1	Entity Relationship . . . . .	9
2	Entity Relationship UMLumlchemaer . . . . .	11
3	Schema Fisico . . . . .	13

## 1 Descrizione del progetto

Il progetto scelto si prefigge di realizzare una base di dati per una competizione di crossfit. Il crossfit, nasce da Greg Glassman negli Stati Uniti all' inizio anni 70. Questa disciplina iniziò a diventare popolare negli anni 90. Nel 1995 nacque la prima palestra box che si trovava a Santa Cruz in California. Il Crossfit è un programma di rafforzamento e condizionamento fisico, pensato per aiutare le persone a conquistare un benessere completo e generale. Tale programma si concentra su una serie di movimenti funzionali che cambiano costantemente, eseguiti ad alta intensità, per raggiungere una prestanza fisica totale e rendere le persone pronte a ogni genere di sfida fisica. Per una competizione di crossfit bisogna avere a disposizione diverse palestre o box. Un box è una palestra dedicata a chi svolge il Crossfit. La differenza tra una palestra e un box si trova nella struttura stessa, in un box si devono avere degli spazi liberi, dove i diversi gruppi possono eseguire esercizi a corpo libero, mentre in una palestra troviamo diversi macchinari che oltre a ridurre lo spazio sono non idonei per gli esercizi di crossfit. In una palestra in genere si allenano diverse persone contemporaneamente, ma con un programma di allenamento totalmente differente tra loro. Nel box gli allenamenti si svolgono tra più gruppi di persone, le quali svolgono lo stesso tipo di allenamento, supervisionati da uno o più Coach. In una competizione di Crossfit siamo in presenza di atleti, che devono essere maggiorenni e in ottima salute per poter partecipare. Un'altra fondamentale caratteristica presente nelle competizioni di crossfit è data dalla partecipazione di atleti in Team. Un Team è composto da un numero variabile di persone, e contiene uno o più Coach. Un Team, per sostenere le spese economiche necessarie alla competizione si avvale di più sponsor che li supportano economicamente. I team come già accennato possiedono uno o più Coach, i quali devono essere in possesso di una o più certificazioni per poter essere definiti Coach. Le certificazioni, sono suddivise in quattro livelli differenti, ognuno identificato da un codice. La competizione prevede che un Team esegua ,un wod, nel box dove sarà valutato da un giudice che gli assegnerà un punteggio compreso tra 0 e 100. Il wod consiste in una lista di esercizi che il team deve eseguire per poter essere valutato. Un wod possiede una data di pubblicazione e una data di termine, entro il quale il team deve completare il wod. La mancata esecuzione di un wod, non determinerà l'esclusione del team dalla competizione, ma solo la conquista di un minor punteggio, utile per il raggiungimento del risultato finale. Un wod deve essere eseguito esclusivamente nel box ad esso associato. Questo darà modo ai diversi team ospitati, di apprezzare le diverse tecniche di allenamento utilizzate in diverse località del mondo. Gli esercizi presenti nel wod devono essere compresi da tutti i Team, per questo motivo

si dovrà disporre di una lista di esercizi con una breve descrizione funzionale. Per quanto riguarda quegli esercizi che comprendono un carico specifico, si devono rispettare determinati vincoli variabili per ogni tipo di attrezzo a seconda degli standard. In ogni competizione si avrà un pubblico. Volendo si potranno memorizzare i dati anagrafici di ogni persona che ha partecipato da spettatore, e si potrà memorizzare in quale box ha fatto da spettatore, per consentire l'utilizzo dei dati da parte dei box per fini statistici. Al termine della competizione sarà decretato il vincitore valutando i risultati ottenuti dai singoli team, per ogni wod, ottenendo una classifica generale.

## 2 Analisi dei requisiti

L'analisi dei requisiti è sicuramente la fase fondamentale per la realizzazione di una base di dati. In questa sezione si raccolgono tutti quei concetti necessari a identificare le entità, e le relazioni che faranno parte dello schema Entity-Relationship. Le informazioni raccolte saranno divise in due tabelle, una per le entità e una per le relazioni.

Tabella 1: Tabella delle entità con le relative descrizioni.

Entità	Descrizione	Attributi	Identificatore
<b>Atleta</b>	Una persona che partecipa alla competizione	E-mail, Nome, Cognome, Data di nascita, Sesso	E-mail
<b>Coach</b>	Una persona che allena un determinato team	E-mail, Nome, Cognome, Data di nascita, Sesso	E-mail
<b>Sponsor</b>	Un'azienda di vario tipo che sponsorizza il team supportandolo economicamente	Nome, Quota versata	Nome
<b>Certificazione</b>	Attesta il grado di preparazione di un Coach	Id-livello, Descrizione	Id-livello
<b>Team</b>	Identifica un insieme di Atleti ai quali si assegnano delle proprietà	Nome-team, Data-iscrizione, Stato	Nome-team
<b>Wod</b>	Raccoglie un insieme di esercizi al quale viene assegnato un nome una data di pubblicazione e scadenza	Id-wod, Data-pubblicazione, Data-termine	Id-wod
<b>Box</b>	Una palestra che si trova in un determinato stato	Nome-box, Stato	Nome-box
<b>Giudice</b>	Identifica una persona che svolgerà il compito di giudice in una competizione	Id-giudice, Nome, Cognome, Data di nascita, Sesso	Id-giudice
<b>Esercizio</b>	Descrive un componente del wod che un atleta deve eseguire in una competizione	Nome-esercizio, Descrizione	Nome-esercizio
<b>Persona</b>	Una persona che farà da pubblico in una o più competizioni	Codice-fiscale, Nome, Cognome, Data di nascita, Sesso	Codice-fiscale

Tabella 2: Tabella delle relazioni con le relative descrizioni.

Relazione	Descrizione	Entità Coinvolte	Attributi
<b>SP_TE</b>	Associa uno sponsor a un team	Sponsor (0,1), Team (1,N)	
<b>AT_TE</b>	Afferisce un atleta a un team	Atleta (1,1), Team (1,N)	
<b>CO_TE</b>	Associa a un team uno o più coach	Coach (1,1), Team (1,N)	
<b>CO_CE</b>	Associa una o più certificazioni a un coach	Coach (1,N), Certificazione (1,N)	
<b>TE_GI_WO</b>	Mette in relazione un team un giudice e un wod per rappresentare una competizione	Team (1,N), Giudice (1,N), Wod (0,N)	Data esecuzione, Punteggio
<b>WO_ES</b>	Determina la composizione di un wod associando a esso uno o più esercizi	Wod (1,N), Esercizio (1,N)	
<b>BO_PE</b>	Associa una persona a uno o più box per rappresentare l'insieme degli spettatori	Box (0,N), Persona (0,N)	Data presenza

### 3 Regole Aziendali

In questa sezione andremo a definire quelle regole che, nella terminologia delle basi di dati relazionali vanno sotto il nome di Regola di Vincolo, e Regola di Derivazione. Queste regole sono molto importanti se si vuole esprimere qualcosa che non può esser espressa tramite lo schema Entity-Relationship.

#### 3.1 Regole di Vincolo

Una regola di vincolo può essere formulata utilizzando il linguaggio naturale, formulando delle asserzioni espresse nella forma dichiarativa.

**RV1** Un atleta deve essere maggiorenne

**RV2** Un coach deve essere maggiorenne

**RV3** Un giudice deve essere maggiorenne

#### 3.2 Regole di Derivazione

Le regole di derivazione possono essere espresse specificando le operazioni (aritmetiche o di altro genere) che permettono di ottenere il concetto derivato.

**RD1** L'età di un atleta,coach,giudice si ottiene effettuando una differenza tra la data corrente e la data di nascita.

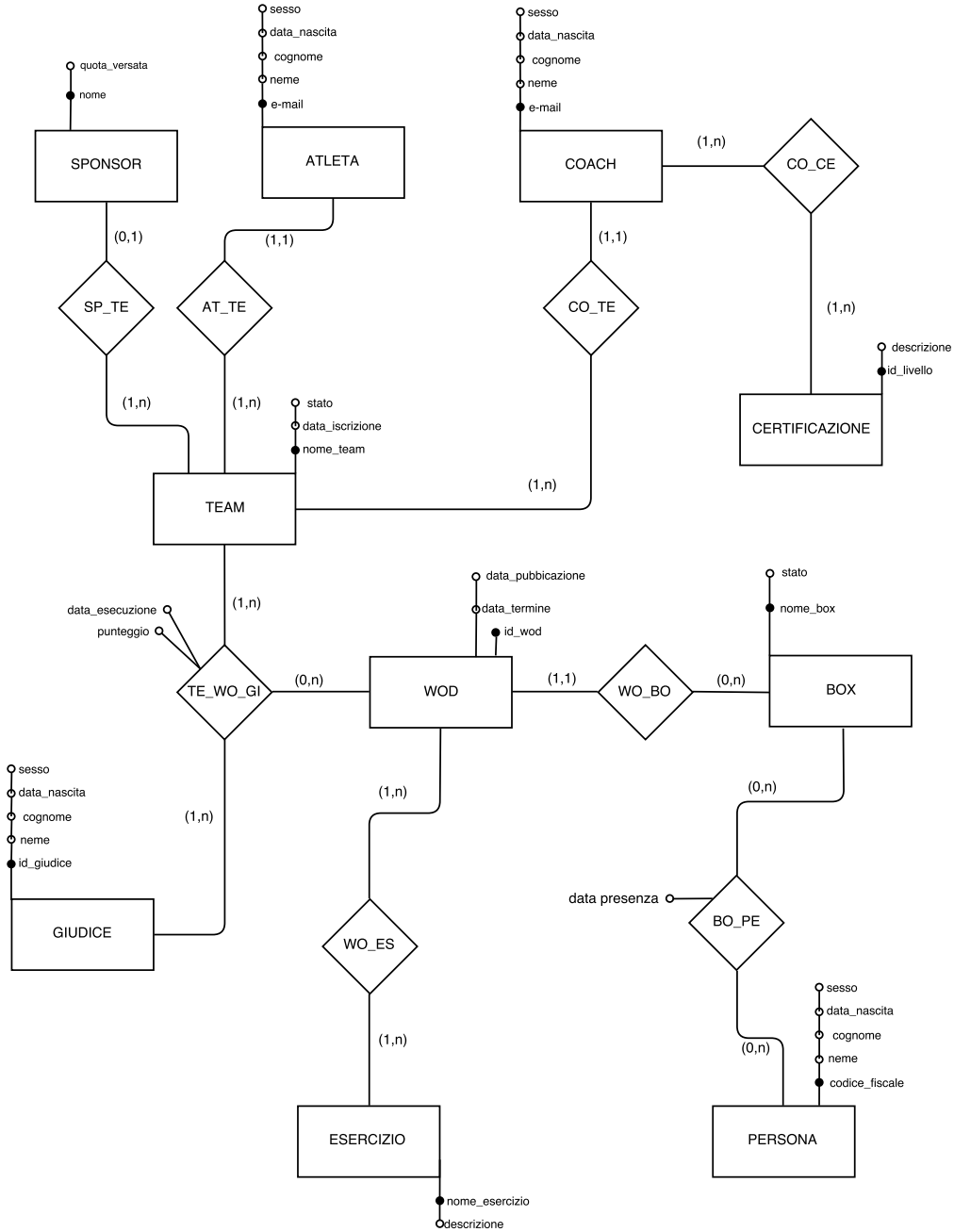
**RD2** Il punteggio finale ottenuto da un team si ottiene effettuando la somma di tutti i punteggi assegnati nei wod eseguiti meno il numero di wod non eseguiti.

### 4 Schema Entity-Relationship

Qui si riporta lo schema Entity-Relationship che descrive la base di dati che si vuole realizzare tramite i costrutti forniti dal modello.

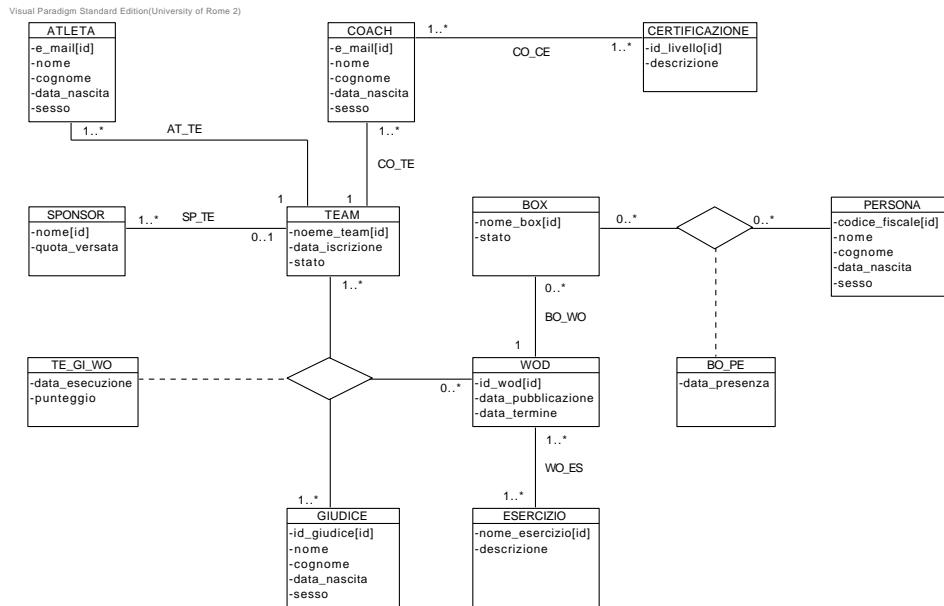


Figura 1: Schema Entity-Relationship.



Normalmente si utilizza una modellazione che comprende tutti i costrutti necessari a rappresentare una base di dati relazionale, ma nella figura successiva possiamo vedere grazie al linguaggio UML (Unified Model Language) come possiamo modellare una base di dati relazionale grazie ai costrutti messi a disposizione da questo linguaggio. L'UML è fortemente utilizzato dall'ingegneria del software per modellare la struttura di un software grazie a un insieme di modelli messi a disposizione dal linguaggio. Nel modellare la base di dati con l'UML si è utilizzato il Class Diagram che normalmente è utilizzato per dare una visione statica degli oggetti che fanno parte di un software.

Figura 2: Schema Entity-Relationship con UML.



## 5 Normalizzazione schema Entity-Relationship

### 5.1 Prima forma normale.

Lo schema si presenta nella prima forma normale dato che tutti gli attributi sono in forma semplice, ovvero indivisibili.

### 5.2 Seconda Forma Normale

Uno schema è nella seconda forma normale, se è in prima forma normale, e tutti gli attributi di uno schema  $R(X)$  sono totalmente dipendenti dalla chiave di  $R(X)$ . Nel caso dello schema presentato ci troviamo in questa situazione. Questo ci permette di affermare che la seconda forma normale è soddisfatta.

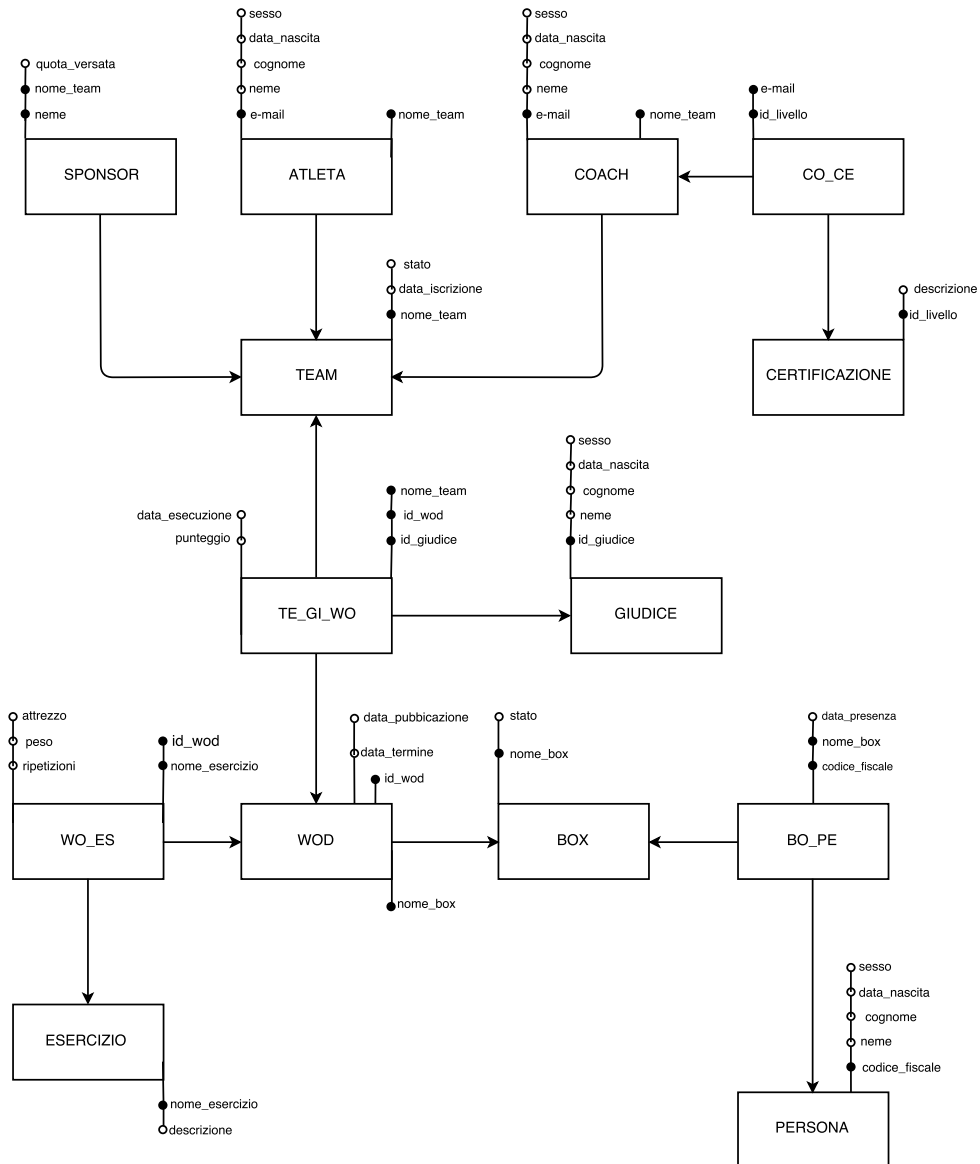
### 5.3 Terza Forma Normale

Lo schema si trova nella terza forma normale dato che ogni attributo che non fa parte della chiave di  $R(X)$  è dipendente in modo non transitivo da ogni chiave.

## 6 Schema Fisico

Qui si presenta lo schema fisico ottenuto dallo schema Entity-Relationship dopo che si è verificato che questo sia nella terza forma normale. Questo schema risulta essere l'ultimo step nella fase progettuale della base di dati, prima che questa prenda vita in un DBMS vero e proprio.

Figura 3: Schema Fisico.



## 7 Creazione delle Entità e delle Relazioni.

Questa parte del progetto mostra gli statements DDL (Data Definition Language). Il DDL è un sottoinsieme del linguaggio SQL che permette di codificare le Entità e le Relazioni presenti nello schema fisico. Il DBMS scelto per realizzare la base di dati è MySQL di Oracle versione 5.6.27.

```
1  /*
2   crea il database se questo non esiste
3  */
4  create database if not exists crossfit2;
5
6  /*
7   seleziona il database crossfit
8  */
9  use crossfit2;
10
11 /*
12 crea la tabella esercizio
13 */
14 create table if not exists esercizio(
15     nome_esercizio VARCHAR(255) NOT NULL,
16     descrizione VARCHAR(255) NOT NULL,
17     primary key (nome_esercizio)
18 ) ENGINE=INNODB;
19
20 /*
21 crea la tabella persona
22 */
23 create table if not exists persona(
24     codice_fiscale VARCHAR(20) NOT NULL,
25     nome VARCHAR(255) NOT NULL,
26     cognome VARCHAR(255) NOT NULL,
27     data_nascita DATE NOT NULL,
28     sesso ENUM('m', 'f'),
29     PRIMARY KEY (codice_fiscale)
30 ) ENGINE=INNODB;
31
32 /*
33 crea la tabella box
34 */
35 create table if not exists box(
36     nome_box VARCHAR(255) NOT NULL,
37     stato VARCHAR(255) NOT NULL,
38     PRIMARY KEY (nome_box)
39 ) ENGINE=INNODB;
40
41 /*
42 crea la tabella BO_PE questa e' una relazione
43 */
44 create table if not exists bo_pe(
45     nome_box VARCHAR(255) NOT NULL,
46     codice_fiscale VARCHAR(255) NOT NULL,
47     data_presenza DATE NOT NULL,
48     FOREIGN KEY (nome_box) REFERENCES box(nome_box),
49     FOREIGN KEY (codice_fiscale) REFERENCES persona(codice_fiscale),
50     PRIMARY KEY (nome_box, codice_fiscale, data_presenza)
51 ) ENGINE=INNODB;
52
53 /*
```

```

54  crea la tabella wod
55  */
56  create table if not exists wod(
57    data_pubblicazione DATE NOT NULL,
58    data_termine DATE NOT NULL,
59    id_wod VARCHAR(255) NOT NULL,
60    nome_box VARCHAR(255) NOT NULL,
61    FOREIGN key (nome_box) REFERENCES box(nome_box) ,
62    PRIMARY KEY (id_wod)
63  ) ENGINE=INNODB;
64
65  /*
66  crea la tabella WO_ES questa e' una relazione
67  */
68  create table if not exists wo_es(
69    id_wod VARCHAR(255) NOT NULL,
70    nome_esercizio VARCHAR(255) NOT NULL,
71    attrezzo VARCHAR(255) NULL,
72    peso INTEGER NULL,
73    ripetizioni INTEGER NOT NULL,
74    FOREIGN KEY (id_wod) REFERENCES wod(id_wod) ,
75    FOREIGN KEY (nome_esercizio) REFERENCES esercizio(nome_esercizio) ,
76    PRIMARY KEY (id_wod,nome_esercizio)
77  ) ENGINE=INNODB;
78
79  /*
80  crea la tabella giudice
81  */
82  create table if not exists giudice(
83    id_giudice VARCHAR(255) NOT NULL,
84    nome VARCHAR(255) NOT NULL,
85    cognome VARCHAR(255) NOT NULL,
86    data_nascita DATE NOT NULL,
87    sesso ENUM('m','f'),
88    PRIMARY KEY (id_giudice)
89  ) ENGINE=INNODB;
90
91  /*
92  crea la tabella team
93  */
94  create table if not exists team(
95    nome VARCHAR(255) NOT NULL,
96    data_iscrizione DATE NOT NULL,
97    stato VARCHAR(255) NOT NULL,
98    PRIMARY KEY (nome)
99  ) ENGINE=INNODB;
100
101  /*
102  crea la tabella sponsor
103  */
104  create table if not exists sponsor(
105    nome VARCHAR(255) NOT NULL,
106    nome_team VARCHAR(255) NOT NULL,
107    quota_versata INTEGER NOT NULL,
108    FOREIGN KEY(nome_team) REFERENCES team(nome) ,
109    PRIMARY KEY(nome)
110  ) ENGINE=INNODB;
111
112  /*
113  crea la tabella team
114  */
115  create table if not exists atleta(

```

```

116 e_mail VARCHAR(255) NOT NULL,
117 nome_team VARCHAR(255) NOT NULL,
118 nome VARCHAR(255) NOT NULL,
119 cognome VARCHAR(255) NOT NULL,
120 data_nascita DATE NOT NULL,
121 sesso ENUM('m', 'f'),
122 FOREIGN KEY (nome_team) REFERENCES team(nome),
123 PRIMARY KEY (e_mail)
124 ) ENGINE=INNODB;
125
126 create table if not exists coach(
127 e_mail VARCHAR(255) NOT NULL,
128 nome_team VARCHAR(255) NOT NULL,
129 nome VARCHAR(255) NOT NULL,
130 cognome VARCHAR(255) NOT NULL,
131 data_nascita DATE NOT NULL,
132 sesso ENUM('m', 'f'),
133 FOREIGN KEY (nome_team) REFERENCES team(nome),
134 PRIMARY KEY (e_mail)
135 ) ENGINE=INNODB;
136
137 /*
138 crea la tabella certificazione
139 */
140 create table if not exists certificazione(
141 idLivello VARCHAR(255) NOT NULL,
142 descrizione VARCHAR(255) NOT NULL,
143 PRIMARY KEY (idLivello)
144 ) ENGINE=INNODB;
145
146 /*
147 crea la tabella CO_CE e' una relazione
148 */
149 create table if not exists co_ce(
150 e_mail VARCHAR(255) NOT NULL,
151 idLivello VARCHAR(255) NOT NULL,
152 FOREIGN KEY (e_mail) REFERENCES coach(e_mail),
153 FOREIGN KEY (idLivello) REFERENCES certificazione(idLivello),
154 PRIMARY KEY (e_mail, idLivello)
155 ) ENGINE=INNODB;
156
157 /*
158 crea la tabella TE_GI_WO e' una relazione
159 */
160 create table if not exists te_gi_wo(
161 id_giudice VARCHAR(255) NOT NULL,
162 id_wod VARCHAR(255) NOT NULL,
163 data_esecuzione DATE NOT NULL,
164 nome_team VARCHAR(255) NOT NULL,
165 punteggio INTEGER NOT NULL,
166 PRIMARY KEY (id_giudice, id_wod, nome_team),
167 UNIQUE (id_wod, nome_team)
168 ) ENGINE=INNODB;

```



## 8 Inserimento dei dati nel database

Si inseriscono dei dati di prova nel database utilizzando gli statements DML (Data Manipulation Language). Il DML è un sottoinsieme del linguaggio SQL.

```
1
2  /*
3   inserimento dei team.
4  */
5
6  insert into team(nome,data_iscrizione,stato) VALUES
7  ("team01","2015-01-20","Italia"),
8  ("team02","2015-02-05","Italia"),
9  ("team03","2015-01-27","Francia"),
10 ("team04","2015-01-27","Germania"),
11 ("team05","2015-01-24","USA"),
12 ("team06","2015-01-24","Regno Unito"),
13 ("team07","2015-02-10","India"),
14 ("team08","2015-01-22","India"),
15 ("team09","2015-01-25","Tailandia"),
16 ("team10","2015-02-09","Russia");
17
18 /*
19 inserimento degli atleti
20 */
21
22 INSERT INTO atleta ( e_mail, nome_team, nome, cognome, data_nascita,
23                     sesso ) VALUES
24 ( 'woodrow@mail.org', 'team01', 'Woodrow', 'Kilbride', '1976-06-03', 'M' ),
25 ( 'jesus@mail.org', 'team01', 'Jesus', 'Hansen', '1991-12-23', 'M' ),
26 ( 'jose@mail.org', 'team01', 'Jose', 'Juriga', '1985-03-06', 'M' ),
27 ( 'antony@mail.org', 'team02', 'Antony', 'Jaeger', '1996-07-02', 'M' ),
28 ( 'micheal@mail.org', 'team02', 'Michael', 'Tetley', '1984-10-16', 'M' ),
29 ( 'delbert@mail.org', 'team02', 'Delbert', 'Starcevic', '1962-08-08', 'M' ),
30 ( 'john@mail.org', 'team03', 'John', 'Balo', '1973-10-04', 'M' ),
31 ( 'myung@mail.org', 'team03', 'Myung', 'Rhett', '1970-10-30', 'M' ),
32 ( 'georgiann@mail.org', 'team03', 'Georgiann', 'Diersen', '1984-04-12', 'M' ),
33 ( 'toni@mail.org', 'team04', 'Toni', 'Monz', '1959-11-21', 'M' ),
34 ( 'jacob@mail.org', 'team04', 'Jacob', 'Padua', '1986-05-26', 'M' ),
35 ( 'kenton@mail.org', 'team04', 'Kenton', 'Sawczynszyn', '1960-01-01', 'M' ),
36 ( 'chris@mail.org', 'team05', 'Chris', 'Eversmann', '1967-11-18', 'M' ),
37 ( 'warren@mail.org', 'team05', 'Warren', 'Dominique', '1986-08-15', 'M' ),
38 ( 'melvin@mail.org', 'team05', 'Melvin', 'Simmons', '1957-08-05', 'M' ),
39 ( 'fredrick@mail.org', 'team06', 'Fredrick', 'Nesting', '1966-04-22', 'M' ),
40 ( 'chester@mail.org', 'team06', 'Chester', 'Andersen', '1956-01-24', 'M' ),
41 ( 'graig@mail.org', 'team06', 'Graig', 'Bonetto', '1974-12-24', 'M' ),
42 ( 'pete@mail.org', 'team07', 'Pete', 'Sharplin', '1976-08-18', 'M' ),
43 ( 'charles@mail.org', 'team07', 'Charles', 'Kepani', '1994-12-15', 'M' ),
44 ( 'eddy@mail.org', 'team07', 'Eddy', 'Schleh', '1961-11-17', 'M' ),
45 ( 'terrance@mail.org', 'team08', 'Terrance', 'Haubert', '1970-05-14', 'M' ),
46 ( 'isaias@mail.org', 'team08', 'Isaias', 'Dieter', '1983-09-03', 'M' ),
47 ( 'dennis@mail.org', 'team08', 'Dennis', 'Dafonseca', '1996-09-27', 'M' ),
48 ( 'beryl@mail.org', 'team09', 'Beryl', 'Lehmkuhl', '1987-06-24', 'M' ),
49 ( 'wilber@mail.org', 'team09', 'Wilber', 'Comp', '1993-10-17', 'M' ),
50 ( 'marlon@mail.org', 'team09', 'Marlon', 'Wikholm', '1981-06-18', 'M' ),
51 ( 'jamaal@mail.org', 'team10', 'Jamaal', 'Giacone', '1968-11-19', 'M' ),
52 ( 'mark@mail.org', 'team10', 'Mark', 'Bernskoetter', '1985-08-17', 'M' ),
```

```

52 ( 'kurtis@mail.org', 'team10', 'Kurtis', 'Innarelli', '1971-05-19', 'M' ),
53 ( 'tifany@mail.org', 'team01', 'Tifany', 'Keziah', '1972-05-13', 'F' ),
54 ( 'raeann@mail.org', 'team01', 'Raeann', 'Lee', '1973-05-17', 'F' ),
55 ( 'jenni@mail.org', 'team01', 'Jenni', 'Nguyen', '1966-04-11', 'F' ),
56 ( 'laraine@mail.org', 'team02', 'Laraine', 'Tenneson', '1977-10-26', 'F' ),
57 ( 'delsie@mail.org', 'team02', 'Delsie', 'Vildosola', '1966-06-01', 'F' ),
58 ( 'sirena@mail.org', 'team02', 'Sirena', 'Stingel', '1960-02-03', 'F' ),
59 ( 'myles@mail.org', 'team03', 'Myles', 'Bonga', '1997-06-06', 'F' ),
60 ( 'marietta@mail.org', 'team03', 'Marietta', 'Vena', '1980-05-02', 'F' ),
61 ( 'beaulah@mail.org', 'team03', 'Beaulah', 'Truslow', '1986-01-24', 'F' ),
62 ( 'jessica@mail.org', 'team04', 'Jessica', 'Slankard', '1970-06-09', 'F' ),
63 ( 'alexandra@mail.org', 'team04', 'Alexandra', 'Hillerud', '1979-05-05', 'F' ),
64 ( 'eunice@mail.org', 'team04', 'Eunice', 'Galle', '1962-04-03', 'F' ),
65 ( 'shirlene@mail.org', 'team05', 'Shirlene', 'Crecco', '1965-06-02', 'F' ),
66 ( 'tenesha@mail.org', 'team05', 'Tenesha', 'Groskreutz', '1977-05-21', 'F' ),
67 ( 'kenyatta@mail.org', 'team05', 'Kenyatta', 'Leuthauser', '1969-05-01', 'F' ),
68 ( 'katharina@mail.org', 'team06', 'Katharina', 'Landefeld', '1987-01-03', 'F' ),
69 ( 'maira@mail.org', 'team06', 'Maira', 'Olay', '1968-12-21', 'F' ),
70 ( 'veronique@mail.org', 'team06', 'Veronique', 'Higashida', '1957-04-11', 'F' ),
71 ( 'latia@mail.org', 'team07', 'Latia', 'Thivierge', '1966-08-29', 'F' ),
72 ( 'gene@mail.org', 'team07', 'Gene', 'Glascoe', '1966-11-07', 'F' ),
73 ( 'joane@mail.org', 'team07', 'Joane', 'Muhlbach', '1979-02-18', 'F' ),
74 ( 'rosie@mail.org', 'team08', 'Rosie', 'Maurais', '1966-02-28', 'F' ),
75 ( 'teodora@mail.org', 'team08', 'Teodora', 'Paalan', '1958-02-08', 'F' ),
76 ( 'camelia@mail.org', 'team08', 'Camelia', 'Gordon', '1992-07-07', 'F' ),
77 ( 'keira@mail.org', 'team09', 'Keira', 'Festa', '1957-07-09', 'F' ),
78 ( 'princess@mail.org', 'team09', 'Princess', 'Cumber', '1977-05-12', 'F' ),
79 ( 'cristina@mail.org', 'team09', 'Cristina', 'Penning', '1966-09-11', 'F' ),
80 ( 'chelsea@mail.org', 'team10', 'Chelsea', 'Sternberger', '1995-06-13', 'F' ),
81 ( 'ivory@mail.org', 'team10', 'Ivory', 'Cepero', '1990-11-30', 'M' ),
82 ( 'adrienne@mail.org', 'team10', 'Adrienne', 'Towley', '1966-12-15', 'F' );
83
84 /*
85 inserimento dei coach
86 */
87
88 INSERT INTO coach ( e_mail, nome_team, nome, cognome, data_nascita,
89 sesso ) VALUES
89 ( 'regena@coach.org', 'team01', 'Regena', 'McFarlen', '1973-03-02', 'F' ),
90 ( 'lakisha@coach.org', 'team02', 'Lakisha', 'Wyze', '1996-08-07', 'F' ),
91 ( 'ronald@coach.org', 'team03', 'Ronald', 'Loparco', '1980-12-20', 'M' ),
92 ( 'janet@coach.org', 'team04', 'Janet', 'Meullion', '1961-08-10', 'F' ),
93 ( 'jonelle@coach.org', 'team05', 'Jonelle', 'Snaples', '1996-09-18', 'F' ),
94 ( 'toshiko@coach.org', 'team06', 'Toshiko', 'Trio', '1986-11-30', 'F' ),
95 ( 'joseph@coach.org', 'team01', 'Joseph', 'Lecy', '1993-02-12', 'F' ),
96 ( 'jeremiah@coach.org', 'team02', 'Jeremiah', 'Parchment', '1965-12-17', 'M' ),
97 ( 'gerardo@coach.org', 'team05', 'Gerardo', 'Halston', '1975-11-21', 'M' ),
98 ( 'joshua@coach.org', 'team07', 'Joshua', 'Ewell', '1979-02-15', 'M' ),
99 ( 'ahmad@coach.org', 'team08', 'Ahmad', 'Lolagne', '1978-07-17', 'M' ),
100 ( 'margie@coach.org', 'team09', 'Margie', 'Pocius', '1966-05-28', 'F' ),
101 ( 'matthew@coach.org', 'team10', 'Matthew', 'Dossantos', '1981-10-29', 'M' ),
102 ( 'shawanda@coach.org', 'team10', 'Shawanda', 'Martinet', '1992-01-14', 'F' ),
103 ( 'megivern@coach.org', 'team06', 'Mary', 'Megivern', '1994-07-31', 'F' );

```

```

104
105
106 /*
107 inserimento dei giudici
108 */
109
110 INSERT INTO giudice ( id_giudice , nome , cognome , data_nascita , sesso )
111 VALUES
112 ( '1cde09e0a' , 'Edwardo' , 'Rafey' , '1971-11-23' , 'M' ) ,
113 ( '6d' , 'James' , 'Arambuia' , '1990-11-02' , 'M' ) ,
114 ( '4766' , 'Jutta' , 'Dezell' , '1978-05-02' , 'F' ) ,
115 ( '3921b1e3' , 'Eric' , 'Kakudji' , '1964-12-30' , 'M' ) ,
116 ( '65aec' , 'Samual' , 'Baugham' , '1972-03-09' , 'M' ) ,
117 ( '69dcb1' , 'Tresa' , 'Dechambeau' , '1994-02-14' , 'F' ) ,
118 ( '13c2' , 'Margaret' , 'Woeste' , '1973-07-31' , 'F' ) ,
119 ( 'eb84d75b' , 'Clayton' , 'Hanstein' , '1964-05-11' , 'M' ) ,
120 ( '49d6c' , 'Marcellus' , 'Batarse' , '1983-05-07' , 'M' ) ,
121 ( '33703cbe3' , 'Exie' , 'Biever' , '1978-05-19' , 'F' ) ,
122 ( '0e5c18' , 'Codi' , 'Baltazar' , '1989-07-08' , 'M' ) ,
123 ( '908151' , 'Patricia' , 'Tomczak' , '1963-12-09' , 'F' ) ,
124 ( 'b4cbeada4a' , 'Eustolia' , 'Balls' , '1972-02-13' , 'F' ) ,
125 ( '7d127d87a' , 'Shonta' , 'Carns' , '1970-07-17' , 'F' ) ,
126 ( '8da432e' , 'Jarvis' , 'Adelmund' , '1980-07-19' , 'M' ) ,
127 ( 'd446e23' , 'Terrence' , 'Aranjo' , '1984-08-07' , 'M' ) ,
128 ( '8ab563d38' , 'Walter' , 'Schuff' , '1965-12-02' , 'M' ) ,
129 ( 'b1c9' , 'Eunice' , 'Naik' , '1987-08-30' , 'F' ) ,
130 ( '5366e7d3b1' , 'Clifton' , 'Geres' , '1987-12-14' , 'M' ) ,
131 ( '4d302dbbc8' , 'Kyle' , 'Kistle' , '1984-08-18' , 'M' ) ,
132 ( 'ad649a' , 'Elvin' , 'Flattery' , '1973-06-02' , 'M' ) ,
133 ( '321a' , 'Mary' , 'Eiland' , '1988-09-23' , 'F' ) ,
134 ( 'c9c9b' , 'Wendolyn' , 'Suddath' , '1959-07-01' , 'F' ) ,
135 ( '7bb581e' , 'Columbus' , 'Lederman' , '1961-01-05' , 'M' ) ;
136
137 /*
138 inserimento dei sponsor
139 */
140
141 INSERT INTO sponsor ( nome , nome_team , quota_versata ) VALUES
142 ( 'rmp_fitness' , 'team01' , 5000 ) ,
143 ( 'junk' , 'team02' , 10000 ) ,
144 ( 'rokfit' , 'team02' , 8000 ) ,
145 ( 'life_aid' , 'team04' , 7900 ) ,
146 ( 'rx' , 'team05' , 12000 ) ,
147 ( 'quest_nutrition' , 'team06' , 5300 ) ,
148 ( 'lurong_living' , 'team07' , 9800 ) ,
149 ( 'crossover_symmetry' , 'team08' , 15000 ) ,
150 ( 'project_x' , 'team09' , 13400 ) ,
151 ( 'skins' , 'team10' , 6200 ) ,
152 ( 'rumbleroller' , 'team01' , 7000 ) ,
153 ( 'caffeine_and_kilos' , 'team02' , 5450 ) ,
154 ( 'reebok' , 'team08' , 5000 ) ,
155 ( 'exosleeve' , 'team04' , 6000 ) ,
156 ( 'affiliateguard' , 'team05' , 8000 ) ,
157 ( 'harbinger' , 'team06' , 20000 ) ;
158
159 /*
160 inserimento delle certificazioni
161 */
162
163 INSERT INTO certificazione ( idlivello , descrizione ) VALUES
164 ( 'cf-11' , "The Level 1 Certificate Course is an introduction to CrossFit
165 's methodology and foundational movements." ) ,

```

```

164 ('cf-12','The Level 2 Certificate Course (formerly the Coach's Prep
      Course) is an intermediate-level seminar that builds on the
      concepts and movements introduced at the Level 1 Certificate Course
      ."),
165 ('cf-13',"The Certified CrossFit Level 3 Trainer has demonstrated
      competency across a breadth of CrossFit material. They understand
      CrossFit methodology and coaching via the Certificate Courses, have
      significant CrossFit coaching experience."),
166 ('cf-14',"The Certified CrossFit Level 4 Coach credential is a one-day
      performance evaluation of a trainer's ability to coach CrossFit
      movements in a small group setting.");

167
168 /*
169 assegnazione delle certificazioni ai coach
170 */
171
172 INSERT INTO co_ce (e_mail, id_livello) VALUES
173 ('ahmad@coach.org','cf-11'),
174 ('ahmad@coach.org','cf-12'),
175 ('gerardo@coach.org','cf-11'),
176 ('gerardo@coach.org','cf-12'),
177 ('gerardo@coach.org','cf-13'),
178 ('janet@coach.org','cf-11'),
179 ('jeremiah@coach.org','cf-11'),
180 ('jonelle@coach.org','cf-11'),
181 ('joseph@coach.org','cf-11'),
182 ('joseph@coach.org','cf-12'),
183 ('joseph@coach.org','cf-13'),
184 ('joseph@coach.org','cf-14'),
185 ('joshua@coach.org','cf-11'),
186 ('joshua@coach.org','cf-12'),
187 ('joshua@coach.org','cf-13'),
188 ('lakisha@coach.org','cf-11'),
189 ('margie@coach.org','cf-11'),
190 ('matthew@coach.org','cf-11'),
191 ('megivern@coach.org','cf-11'),
192 ('regena@coach.org','cf-11'),
193 ('ronald@coach.org','cf-11'),
194 ('shawanda@coach.org','cf-11'),
195 ('toshiko@coach.org','cf-11'),
196 ('toshiko@coach.org','cf-12');
197
198 /*
199 inserimento degli esercizi
200 */
201
202 INSERT INTO esercizio (nome_esercizio, descrizione) VALUES
203 ('chest-to-bar pull-up','aa'),
204 ('power clean','bb'),
205 ('squat','cc'),
206 ('run','dd'),
207 ('thruster','ee'),
208 ('ghd sit-up','ff'),
209 ('pull-up','gg'),
210 ('hip extension','hh'),
211 ('hang squat clean','ii'),
212 ('one-legged squat','jj'),
213 ('double-under','kk'),
214 ('behind-the-neck jerk','ll'),
215 ('row','mm'),
216 ('toest-tobar','nn'),
217 ('deadlift','oo'),

```

```
218 ('handstand push-up', 'pp'),
219 ('box jump', 'qq'),
220 ('front squat', 'ss'),
221 ('body-weight deadlift', 'tt'),
222 ('body-weight bench press', 'uu'),
223 ('body-weight clean', 'vv'),
224 ('kettlebell swing', 'ww'),
225 ('burpress', 'xx'),
226 ('woll-ball shot', 'yy'),
227 ('squat snatch', 'zz'),
228 ('bike', 'aaa'),
229 ('back squat', 'bbb'),
230 ('push press', 'ccc'),
231 ('hang power clean', 'ddd'),
232 ('muscle-up', 'eee'),
233 ('shoulder-to-overhead', 'fff'),
234 ('snatch', 'ggg'),
235 ('clean and jerk', 'hhh'),
236 ('split jerk', 'iii'),
237 ('bar-facing burpress', 'jjj'),
238 ('hang squat snatch', 'kkk');
239
240 /*
241 inserimento dei box
242 */
243
244 INSERT INTO box (nome_box, stato) VALUES
245 ('crossfit sigonella', 'italy'),
246 ('crossfit vicenza', 'italy'),
247 ('crossfit aviano', 'italy'),
248 ('reebok crossfit alcamo', 'italy'),
249 ('crossfit stade des alpes', 'france'),
250 ('crossfit nancy', 'france'),
251 ('crossfit montpellier', 'france'),
252 ('crossfit essen', 'germany'),
253 ('myleo crossfit', 'germany'),
254 ('crossfit am main', 'germany'),
255 ('crossfit hh', 'germany'),
256 ('crossfit mildenhall', 'united kingdom'),
257 ('crossfit air command', 'united kingdom'),
258 ('crossfit keelhaul', 'united kingdom'),
259 ('crossfit pirbright', 'united kingdom'),
260 ('crossfit conz', 'united kingdom');
261
262 /*
263 inserimento dei wod
264 */
265
266 INSERT INTO wod (data_pubblicazione, data_termine, id_wod, nome_box)
267 VALUES
268 ('2015-01-15', '2015-01-20', '2rfupf2bhl', 'crossfit sigonella'),
269 ('2015-01-21', '2015-01-28', 'mwhmcjajq0', 'crossfit vicenza'),
270 ('2015-02-02', '2015-02-20', 'dmposmt100', 'crossfit stade des alpes'),
271 ('2015-04-10', '2015-04-16', 'in8ntkmt5a', 'crossfit essen'),
272 ('2015-05-05', '2015-05-18', '9gtv66fwvr', 'myleo crossfit'),
273 ('2015-05-22', '2015-05-26', '7mlive554u', 'myleo crossfit'),
274 ('2015-07-01', '2015-07-10', 'fdq0kwf00m', 'crossfit mildenhall'),
275 ('2015-07-12', '2015-07-15', 'rawrnse1h2', 'crossfit air command'),
276 ('2015-07-25', '2015-07-29', 'a2avztnh3u', 'crossfit air command'),
277 ('2015-08-13', '2015-08-22', '4jjtfkklw8', 'crossfit conz'),
278 ('2015-08-24', '2015-08-29', 'obs7puksk9', 'crossfit pirbright');
```

```

279 /*
280 inserimento composizione wod
281 */
282 INSERT INTO wo_es (id_wod, nome_esercizio, attrezzo, peso, ripetizioni)
    VALUES
283 ('2rfupf2bhl', 'chest-to-bar pull-up', 'sbarra', NULL, 9),
284 ('2rfupf2bhl', 'power clean', 'bilancere', 135, 15),
285 ('2rfupf2bhl', 'squat', NULL, NULL, 21),
286 ('2rfupf2bhl', 'run', NULL, NULL, 400),
287 ('2rfupf2bhl', 'wooll-ball shot', 'mad ball', 20, 50),
288 ('mwhmcjajq0', 'ghd sit-up', null, null, 30),
289 ('mwhmcjajq0', 'run', null, 45, 100),
290 ('mwhmcjajq0', 'front squat', null, null, 40),
291 ('dmposmt100', 'squat', null, null, 150),
292 ('dmposmt100', 'hip extension', null, null, 75),
293 ('dmposmt100', 'pull-up', 'sbarra', null, 35),
294 ('dmposmt100', 'burpess', null, null, 120),
295 ('in8ntkmt5a', 'double-under', 'corda', null, 45),
296 ('in8ntkmt5a', 'hang squat snatch', 'bilancere', 150, 10),
297 ('9gtv66fwvr', 'row', 'row machine', null, 250),
298 ('9gtv66fwvr', 'deadlift', 'bilancere', 45, 15),
299 ('9gtv66fwvr', 'handstand push-up', null, null, 20),
300 ('7mlive554u', 'box jump', 'box', null, 15),
301 ('7mlive554u', 'kettlebell swing', 'kettlebell', 40, 15),
302 ('7mlive554u', 'muscle-up', 'sbarra', null, 25),
303 ('7mlive554u', 'clean and jerk', 'bilancere', 45, 13),
304 ('fdq0kwf00m', 'squat snatch', 'bilancere', 30, 10),
305 ('fdq0kwf00m', 'one-legged squat', null, null, 20),
306 ('rawrnselh2', 'bar-facing burpess', 'bilancere', null, 17),
307 ('rawrnselh2', 'bike', 'bike', null, 600),
308 ('rawrnselh2', 'push press', 'bilancere', 55, 8),
309 ('rawrnselh2', 'hang squat clean', 'bilancere', 15, 20),
310 ('rawrnselh2', 'shoulder-to-overhead', 'bilancere', 30, 20),
311 ('a2avztnh3u', 'split jerk', 'bilancere', 200, 10),
312 ('a2avztnh3u', 'hang squat snatch', 'bilancere', 70, 13),
313 ('a2avztnh3u', 'one-legged squat', null, null, 60),
314 ('4jttfkkkw8', 'run', null, null, 900),
315 ('4jttfkkkw8', 'double-under', 'corda', null, 150),
316 ('4jttfkkkw8', 'squat', null, null, 150),
317 ('4jttfkkkw8', 'row', 'row machine', null, 500),
318 ('obs7puksk9', 'kettlebell swing', 'kettlebell', 15, 25),
319 ('obs7puksk9', 'pull-up', 'sbarra', null, 25),
320 ('obs7puksk9', 'push press', 'bilancere', 55, 40),
321 ('obs7puksk9', 'back squat', 'bilancere', 60, 45),
322 ('obs7puksk9', 'power clean', 'bilancere', 30, 29),
323 ('obs7puksk9', 'burpess', null, null, 200);
324
325 /*
326 inserimento delle competizioni
327 */
328
329 INSERT INTO te_gi_wo (id_giudice, id_wod, data_esecuzione, nome_team,
    punteggio) VALUES
330 ('0e5c18', '7mlive554u', '2015-05-25', 'team07', 54),
331 ('0e5c18', 'mwhmcjajq0', '2015-01-21', 'team07', 10),
332 ('0e5c18', 'obs7puksk9', '2015-08-27', 'team06', 71),
333 ('0e5c18', 'rawrnselh2', '2015-07-14', 'team08', 75),
334 ('13c2', '4jttfkkkw8', '2015-08-13', 'team04', 90),
335 ('13c2', '7mlive554u', '2015-05-24', 'team10', 5),
336 ('13c2', 'a2avztnh3u', '2015-07-25', 'team06', 47),
337 ('13c2', 'fdq0kwf00m', '2015-07-04', 'team06', 44),
338 ('1cde09e0a', 'rawrnselh2', '2015-07-12', 'team10', 62),

```

```

339 ('321a', '4jttfkkkw8', '2015-08-19', 'team08', 49),
340 ('321a', 'dmposmt100', '2015-02-10', 'team01', 61),
341 ('33703cbe3', 'fdq0kwf00m', '2015-07-02', 'team07', 77),
342 ('33703cbe3', 'in8ntkmt5a', '2015-04-12', 'team01', 71),
343 ('33703cbe3', 'in8ntkmt5a', '2015-04-12', 'team09', 23),
344 ('3921b1e3', 'dmposmt100', '2015-02-07', 'team07', 20),
345 ('3921b1e3', 'obs7puksk9', '2015-08-25', 'team02', 74),
346 ('4766', '9gtv66fwvr', '2015-05-05', 'team05', 27),
347 ('49d6c', 'in8ntkmt5a', '2015-04-12', 'team06', 2),
348 ('49d6c', 'mwhmcjajq0', '2015-01-23', 'team04', 56),
349 ('4d302dbbc8', '2rfupf2bhl', '2015-01-15', 'team01', 45),
350 ('4d302dbbc8', 'in8ntkmt5a', '2015-04-13', 'team04', 8),
351 ('4d302dbbc8', 'obs7puksk9', '2015-08-26', 'team01', 30),
352 ('4d302dbbc8', 'obs7puksk9', '2015-08-24', 'team09', 54),
353 ('5366e7d3b1', '7mlive554u', '2015-05-25', 'team09', 91),
354 ('5366e7d3b1', 'a2avztnh3u', '2015-07-26', 'team03', 84),
355 ('5366e7d3b1', 'mwhmcjajq0', '2015-01-27', 'team05', 7),
356 ('5366e7d3b1', 'rawrnslh2', '2015-07-14', 'team06', 85),
357 ('65aec', '2rfupf2bhl', '2015-01-16', 'team10', 66),
358 ('69dcb1', 'dmposmt100', '2015-02-14', 'team03', 96),
359 ('6d', '7mlive554u', '2015-05-24', 'team03', 42),
360 ('6d', '7mlive554u', '2015-05-24', 'team05', 82),
361 ('6d', 'dmposmt100', '2015-02-02', 'team04', 59),
362 ('7bb581e', 'dmposmt100', '2015-02-03', 'team06', 6),
363 ('7bb581e', 'dmposmt100', '2015-02-11', 'team10', 95),
364 ('7bb581e', 'fdq0kwf00m', '2015-07-08', 'team02', 29),
365 ('7bb581e', 'obs7puksk9', '2015-08-28', 'team05', 41),
366 ('7bb581e', 'rawrnslh2', '2015-07-13', 'team03', 51),
367 ('7d127d87a', 'dmposmt100', '2015-02-14', 'team02', 36),
368 ('8da432e', '2rfupf2bhl', '2015-01-16', 'team02', 63),
369 ('8da432e', '2rfupf2bhl', '2015-01-15', 'team08', 66),
370 ('8da432e', '7mlive554u', '2015-05-23', 'team06', 92),
371 ('8da432e', 'dmposmt100', '2015-02-08', 'team08', 59),
372 ('8da432e', 'fdq0kwf00m', '2015-07-07', 'team09', 33),
373 ('908151', '7mlive554u', '2015-05-24', 'team08', 28),
374 ('908151', 'in8ntkmt5a', '2015-04-14', 'team02', 27),
375 ('908151', 'in8ntkmt5a', '2015-04-14', 'team03', 53),
376 ('908151', 'mwhmcjajq0', '2015-01-27', 'team09', 81),
377 ('908151', 'obs7puksk9', '2015-08-27', 'team04', 16),
378 ('908151', 'rawrnslh2', '2015-07-13', 'team09', 16),
379 ('ad649a', '4jttfkkkw8', '2015-08-14', 'team03', 22),
380 ('ad649a', '4jttfkkkw8', '2015-08-21', 'team06', 62),
381 ('b1c9', '9gtv66fwvr', '2015-05-10', 'team04', 4),
382 ('b1c9', '9gtv66fwvr', '2015-05-13', 'team06', 8),
383 ('b1c9', 'a2avztnh3u', '2015-07-26', 'team09', 21),
384 ('b1c9', 'a2avztnh3u', '2015-07-25', 'team10', 11),
385 ('b1c9', 'dmposmt100', '2015-02-14', 'team09', 38),
386 ('b4cbeada4a', 'obs7puksk9', '2015-08-26', 'team07', 59),
387 ('b4cbeada4a', 'obs7puksk9', '2015-08-24', 'team08', 29),
388 ('c9c9b', '9gtv66fwvr', '2015-05-12', 'team01', 75),
389 ('c9c9b', 'fdq0kwf00m', '2015-07-02', 'team03', 94),
390 ('c9c9b', 'mwhmcjajq0', '2015-01-22', 'team06', 7),
391 ('c9c9b', 'rawrnslh2', '2015-07-14', 'team02', 58),
392 ('eb84d75b', '2rfupf2bhl', '2015-01-16', 'team04', 18),
393 ('eb84d75b', '9gtv66fwvr', '2015-05-13', 'team08', 93),
394 ('eb84d75b', 'a2avztnh3u', '2015-07-25', 'team04', 76),
395 ('eb84d75b', 'mwhmcjajq0', '2015-01-22', 'team02', 1),
396 ('eb84d75b', 'rawrnslh2', '2015-07-13', 'team07', 16);
397
398 /*
399 inserimento persona spettatore
400 */

```



```

401
402 insert into persona (codice_fiscale, nome, cognome, data_nascita, sesso
    ) values
403 ('580-01-8340', 'Lisa', 'Banks', '1996-03-07', 'F'),
404 ('664-82-2845', 'Angela', 'Garza', '1991-03-20', 'F'),
405 ('326-27-1686', 'Judith', 'Lynch', '1975-07-09', 'F'),
406 ('816-67-2964', 'Kathleen', 'Cole', '1967-12-08', 'F'),
407 ('781-81-5500', 'Carolyn', 'Ortiz', '1987-12-17', 'F'),
408 ('944-22-7260', 'Robin', 'Stephens', '1976-07-01', 'F'),
409 ('602-33-7537', 'Shirley', 'Reid', '1984-11-25', 'F'),
410 ('620-94-6245', 'Irene', 'Black', '1984-03-27', 'F'),
411 ('532-64-7942', 'Jessica', 'Reynolds', '1966-09-30', 'F'),
412 ('479-24-6079', 'Susan', 'Harris', '1996-03-25', 'F'),
413 ('764-00-2718', 'Frances', 'Tucker', '1968-10-25', 'F'),
414 ('690-77-0593', 'Kathy', 'Harris', '1992-03-03', 'F'),
415 ('298-10-5658', 'Frances', 'Walker', '1971-01-07', 'F'),
416 ('115-89-3173', 'Jacqueline', 'Medina', '1981-03-16', 'F'),
417 ('529-53-3240', 'Julia', 'Daniels', '1965-10-12', 'F'),
418 ('943-58-4348', 'Gloria', 'Fields', '1986-05-12', 'F'),
419 ('865-08-1443', 'Sharon', 'Diaz', '1956-03-17', 'F'),
420 ('342-83-2663', 'Joan', 'Snyder', '1974-07-14', 'F'),
421 ('261-81-9687', 'Jessica', 'Duncan', '1980-07-24', 'F'),
422 ('277-40-0724', 'Jennifer', 'Simmons', '1985-02-04', 'F'),
423 ('343-31-9575', 'Carol', 'Barnes', '1958-03-11', 'F'),
424 ('409-55-4096', 'Brenda', 'Peters', '1974-07-15', 'F'),
425 ('135-00-3307', 'Eugene', 'Roberts', '1989-05-29', 'M'),
426 ('136-56-4598', 'Victor', 'Matthews', '1956-10-17', 'M'),
427 ('828-40-8126', 'Justin', 'King', '1988-07-27', 'M'),
428 ('532-22-7506', 'Kevin', 'Thomas', '1991-01-14', 'M'),
429 ('820-12-4770', 'Douglas', 'Alexander', '1978-05-09', 'M'),
430 ('744-38-2229', 'Shawn', 'Hill', '1971-06-04', 'M'),
431 ('604-42-6150', 'Patrick', 'White', '1985-04-10', 'M'),
432 ('393-88-8135', 'Matthew', 'Fields', '1975-05-25', 'M'),
433 ('980-45-6742', 'Louis', 'Simpson', '1978-09-22', 'M'),
434 ('280-16-3253', 'George', 'Warren', '1961-10-06', 'M'),
435 ('586-39-5111', 'Anthony', 'Lynch', '1977-11-22', 'M'),
436 ('117-69-2861', 'Anthony', 'Armstrong', '1961-10-03', 'M'),
437 ('782-20-0523', 'Harry', 'Mcdonald', '1993-03-16', 'M'),
438 ('720-88-5936', 'Joshua', 'Frazier', '1966-01-12', 'M'),
439 ('116-41-8799', 'Carl', 'Myers', '1970-08-26', 'M'),
440 ('888-29-8773', 'Jesse', 'Baker', '1986-02-01', 'M'),
441 ('667-15-3793', 'Jack', 'Brown', '1995-01-28', 'M'),
442 ('331-57-8528', 'Peter', 'Moreno', '1988-01-31', 'M'),
443 ('293-02-7297', 'Louis', 'Powell', '1959-09-27', 'M'),
444 ('781-94-4378', 'Larry', 'King', '1963-09-13', 'M'),
445 ('140-63-9038', 'Ryan', 'Gomez', '1983-07-27', 'M');
446
447 /*
448 inserimento degli spettatori
449 */
450
451 INSERT INTO bo_pe (nome_box, codice_fiscale, data_presenza) VALUES
452 ('myleo crossfit', '580-01-8340', '2015-05-25'),
453 ('myleo crossfit', '664-82-2845', '2015-05-25'),
454 ('myleo crossfit', '326-27-1686', '2015-05-25'),
455 ('myleo crossfit', '816-67-2964', '2015-05-25'),
456 ('myleo crossfit', '781-81-5500', '2015-05-25'),
457 ('myleo crossfit', '135-00-3307', '2015-05-25'),
458 ('myleo crossfit', '136-56-4598', '2015-05-25'),
459 ('crossfit sigonella', '580-01-8340', '2015-01-15'),
460 ('crossfit sigonella', '944-22-7260', '2015-01-15'),
461 ('crossfit sigonella', '828-40-8126', '2015-01-15'),

```



```
462 ('crossfit sigonella', '532-22-7506', '2015-01-15'),
463 ('crossfit sigonella', '602-33-7537', '2015-01-16'),
464 ('crossfit sigonella', '620-94-6245', '2015-01-16'),
465 ('crossfit sigonella', '532-64-7942', '2015-01-16'),
466 ('crossfit sigonella', '820-12-4770', '2015-01-16'),
467 ('crossfit vicenza', '479-24-6079', '2015-01-21'),
468 ('crossfit vicenza', '764-00-2718', '2015-01-21'),
469 ('crossfit vicenza', '690-77-0593', '2015-01-21'),
470 ('crossfit vicenza', '298-10-5658', '2015-01-21'),
471 ('crossfit vicenza', '136-56-4598', '2015-01-22'),
472 ('crossfit vicenza', '532-22-7506', '2015-01-22'),
473 ('crossfit vicenza', '744-38-2229', '2015-01-23'),
474 ('crossfit vicenza', '828-40-8126', '2015-01-27'),
475 ('crossfit vicenza', '326-27-1686', '2015-01-27'),
476 ('crossfit vicenza', '580-01-8340', '2015-01-27'),
477 ('crossfit vicenza', '532-64-7942', '2015-01-27'),
478 ('crossfit vicenza', '744-38-2229', '2015-01-27'),
479 ('crossfit stade des alpes', '664-82-2845', '2015-02-02'),
480 ('crossfit stade des alpes', '690-77-0593', '2015-02-02'),
481 ('crossfit stade des alpes', '604-42-6150', '2015-02-14'),
482 ('crossfit stade des alpes', '393-88-8135', '2015-02-14'),
483 ('crossfit stade des alpes', '980-45-6742', '2015-02-14'),
484 ('crossfit stade des alpes', '280-16-3253', '2015-02-14'),
485 ('crossfit stade des alpes', '586-39-5111', '2015-02-14'),
486 ('crossfit stade des alpes', '115-89-3173', '2015-02-14'),
487 ('crossfit essen', '115-89-3173', '2015-04-13'),
488 ('crossfit essen', '620-94-6245', '2015-04-13'),
489 ('myleo crossfit', '529-53-3240', '2015-05-13'),
490 ('myleo crossfit', '943-58-4348', '2015-05-13'),
491 ('myleo crossfit', '865-08-1443', '2015-05-13'),
492 ('myleo crossfit', '342-83-2663', '2015-05-13'),
493 ('crossfit mildenhall', '261-81-9687', '2015-07-02'),
494 ('crossfit mildenhall', '117-69-2861', '2015-07-02'),
495 ('crossfit mildenhall', '529-53-3240', '2015-07-02'),
496 ('crossfit mildenhall', '604-42-6150', '2015-07-07'),
497 ('crossfit air command', '479-24-6079', '2015-07-12'),
498 ('crossfit air command', '764-00-2718', '2015-07-12'),
499 ('crossfit air command', '115-89-3173', '2015-07-12'),
500 ('crossfit air command', '529-53-3240', '2015-07-12'),
501 ('crossfit air command', '115-89-3173', '2015-07-26'),
502 ('crossfit air command', '865-08-1443', '2015-07-26'),
503 ('crossfit air command', '744-38-2229', '2015-07-26'),
504 ('crossfit air command', '136-56-4598', '2015-07-26'),
505 ('crossfit air command', '690-77-0593', '2015-07-26'),
506 ('crossfit air command', '816-67-2964', '2015-07-26'),
507 ('crossfit air command', '782-20-0523', '2015-07-26'),
508 ('crossfit conz', '720-88-5936', '2015-08-19'),
509 ('crossfit conz', '116-41-8799', '2015-08-19'),
510 ('crossfit conz', '277-40-0724', '2015-08-19'),
511 ('crossfit conz', '343-31-9575', '2015-08-19'),
512 ('crossfit conz', '409-55-4096', '2015-08-19'),
513 ('crossfit conz', '529-53-3240', '2015-08-19'),
514 ('crossfit conz', '580-01-8340', '2015-08-19'),
515 ('crossfit conz', '602-33-7537', '2015-08-19'),
516 ('crossfit pirbright', '342-83-2663', '2015-08-26'),
517 ('crossfit pirbright', '479-24-6079', '2015-08-26'),
518 ('crossfit pirbright', '888-29-8773', '2015-08-28'),
519 ('crossfit pirbright', '667-15-3793', '2015-08-28'),
520 ('crossfit pirbright', '331-57-8528', '2015-08-28'),
521 ('crossfit pirbright', '293-02-7297', '2015-08-28'),
522 ('crossfit pirbright', '781-94-4378', '2015-08-28'),
523 ('crossfit pirbright', '140-63-9038', '2015-08-28');
```

## 9 Query

In questa sezione si elencano diverse query d'interesse relative alla base di dati che è stata realizzata precedentemente.

### 9.1 Query 1

```
1  /*
2  mostra_nome, cognome e le certificazioni possedute dai coach
3  */
4  select t.nome,t.cognome,c1.id_livello as 'certif. livel 1',c2.
       id_livello as 'certif. livel 2',
5  c3.id_livello as 'certif. livel 3',c4.id_livello as 'certif. livel 4'
6  from coach as t LEFT OUTER JOIN co_ce as c1
7  on t.e_mail=c1.e_mail and c1.id_livello='cf-11 '
8  LEFT OUTER JOIN co_ce AS c2
9  on t.e_mail=c2.e_mail and c2.id_livello='cf-12 '
10 LEFT OUTER join co_ce AS c3
11 ON t.e_mail=c3.e_mail and c3.id_livello='cf-13 '
12 LEFT OUTER JOIN co_ce AS c4
13 ON t.e_mail=c4.e_mail AND c4.id_livello='cf-14 ';
```

### 9.2 Query 2

```
1  /*
2  vogliamo sapere nome, cognome, sesso dei coach di un certo team
3  */
4  SELECT nome,cognome,sesso
5  FROM coach
6  WHERE nome_team='team01';
```

### 9.3 Query 3

```
1  /*
2  vogliamo sapere quanti sponsor possiedono i team e la somma delle quote
       investite dai sponsor per questi team.
3  */
4  SELECT nome_team AS 'Nome team', count(nome) AS 'Numero sponsor', sum(
       quota_versata) AS 'Budget Totale'
5  FROM sponsor
6  GROUP BY nome_team;
```

## 9.4 Query 4

```
1  /*
2  vogliamo sapere quali stati si sono registrati nella competizione e
   per ogni stato vogliamo sapere quanti gruppi
3  lo rappresentano e la media dell'eta' che li compone.
4  */
5  SELECT num_state.stato,num_state.'team per stato',avg_age.'eta' media '
6  FROM (select stato,count(stato) as 'team per stato'
7        from team group by stato) AS num_state
8  INNER JOIN
9        (SELECT stato , avg(date_format(from_days(datediff(now()),atleta.
        data_nascita)),'%Y')+0) AS 'eta' media '
10 FROM team,atleta
11 WHERE atleta.nome_team=team.nome group by team.stato) AS avg_age
12 ON num_state.stato=avg_age.stato;
```

## 9.5 Query 5

```
1  /*
2  vogliamo determinare la media delle quote investite dagli sponsor e in
   base alla media vogliamo determinare
3  quali team si sono aggiudicati i sponsor che hanno investito con una
   quota maggiore o uguale alla media e
4  quale stato rappresenta il team
5  */
6  SELECT nome_team AS 'Nome team', quota_versata AS 'Budget', t.stato
7  FROM (select avg(quota_versata) AS 'media' from sponsor) AS sm INNER
8        JOIN sponsor AS st
9  ON st.quota_versata >= sm.media
10 INNER JOIN team AS t
11 ON st.nome_team = t.nome;
```

## 9.6 Query 6

```
1  /*
2  seleziona il numero complessivo di spettatori per ogni box per avere l'
   affluenza avuta nella struttura.
3  */
4  select nome_box as 'Nome box' ,count(nome_box) as 'Spettatori'
5  FROM bo_pe
6  GROUP BY nome_box;
```

## 9.7 Query 7

```
1  /*
2  vogliamo determinare la classifica parziale di un determinato wod per
   un certo stato e vogliamo anche sapere
3  il nome e il cognome del giudice che ha assegnato il punteggio.
4  */
5  SELECT te_gi_wo.nome_team,te_gi_wo.data_esecuzione,te_gi_wo.id_wod,
        te_gi_wo.punteggio,giudice.nome,giudice.cognome,team.stato
6  FROM te_gi_wo INNER JOIN team
7  ON te_gi_wo.nome_team = team.nome
8  INNER JOIN giudice
9  ON te_gi_wo.id_giudice = giudice.id_giudice
10 WHERE team.stato = 'italy' AND te_gi_wo.id_wod='2rfupf2bhl' ORDER BY
    te_gi_wo.punteggio DESC;
```

## 9.8 Query 8

```
1  /*
2  come la query precedente ma vogliamo ottenere il risultato per ogni
   stato che ha partecipa al wod specifico
3  */
4  SELECT te_gi_wo.nome_team,te_gi_wo.data_esecuzione,te_gi_wo.id_wod,
        te_gi_wo.punteggio,giudice.nome,giudice.cognome,team.stato
5  FROM te_gi_wo INNER JOIN team
6  ON te_gi_wo.nome_team = team.nome
7  INNER JOIN giudice
8  ON te_gi_wo.id_giudice = giudice.id_giudice
9  WHERE te_gi_wo.id_wod='2rfupf2bhl' ORDER BY te_gi_wo.punteggio DESC;
```

## 9.9 Query 9

```
1  /*
2  vogliamo ottenere una classifica suddivisa per wod e per ogni wod
   vogliamo ricavare il nome del team la data di esecuzione
3  il punteggio ottenuto nome e cognome del giudice e lo stato che
   rappresenta il team.
4  */
5  SELECT te_gi_wo.id_wod,te_gi_wo.nome_team,te_gi_wo.data_esecuzione,
        te_gi_wo.punteggio,giudice.nome,giudice.cognome,team.stato
6  FROM te_gi_wo INNER JOIN team
7  ON te_gi_wo.nome_team = team.nome
8  INNER JOIN giudice
9  ON te_gi_wo.id_giudice = giudice.id_giudice
10 INNER JOIN wod
11 ON te_gi_wo.id_wod=wod.id_wod ORDER BY te_gi_wo.id_wod,te_gi_wo.
    punteggio DESC;
```

## 9.10 Query 10

```

1  /*
2  vogliamo avere la classifica generale al termine della competizione e i
   dati che vogliamo ottenere sono il nome del team il numero di wod
3  eseguiti e il numero di wod mancanti il punteggio realizzato sara
   calcolato in base alla somma dei punteggi ottenuti nei singoli wod
   meno
4  il numero di wod non eseguiti, infine vogliamo sapere di che stato fa
   parte il team.
5  */
6  SELECT te_gi_wo.nome_team,count(id_wod) AS 'Wod eseguiti',num_wod.nw-
   count(id_wod) AS 'Wod mancanti',
7  sum(punteggio)-(num_wod.nw-count(id_wod)) AS 'Punteggio',team.stato
8  FROM te_gi_wo INNER JOIN team
9  ON te_gi_wo.nome_team=team.nome
10 INNER JOIN (SELECT count(id_wod) AS nw FROM wod) AS num_wod
11 GROUP BY te_gi_wo.nome_team ORDER BY 'Punteggio' DESC;

```

## 9.11 Query 11

```

1  /*
2  dato il nome e il cognome di un giudice vogliamo sapere quali team ha
   giudicato in che data e che punteggio gli e' stato
3  assegnato e il nome del box in cui si e' eseguito il wod.
4  */
5  SELECT te_gi_wo.nome_team,te_gi_wo.data_esecuzione,te_gi_wo.punteggio,
   wod.nome_box
6  FROM giudice INNER JOIN te_gi_wo
7  ON giudice.id_giudice=te_gi_wo.id_giudice
8  INNER JOIN wod
9  ON te_gi_wo.id_wod=wod.id_wod
10 WHERE giudice.nome='Jutta' AND giudice.cognome='Dezell';

```

## 9.12 Query 12

```

1  /*
2  vogliamo selezionare il nome dei team il nome degli sponsor a loro
   associati, nome e cognome del giudice che li ha giudicati e il box
   dove
3  hanno eseguito il wod. Di questi vogliamo solo i dati di quei team che
   hanno due coach di sesso femminile e che la data di nascita dei
4  giudici siano coprese in un certo intervallo di date e che siano di
   sesso femminile
5  */
6  SELECT te_gi_wo.id_wod,te_gi_wo.nome_team,sponsor.nome AS 'sponsor',
   giudice.nome,giudice.cognome,giudice.data_nascita,wod.nome_box
7  FROM (select nome_team,count(*) as 'nc' from coach where sesso='f'
   group by nome_team having 'nc'=2) AS tbl
8  INNER JOIN te_gi_wo
9  ON tbl.nome_team=te_gi_wo.nome_team
10 INNER JOIN giudice
11 ON te_gi_wo.id_giudice=giudice.id_giudice
12 INNER JOIN sponsor
13 ON sponsor.nome_team=te_gi_wo.nome_team
14 INNER JOIN wod
15 ON te_gi_wo.id_wod=wod.id_wod
16 WHERE giudice.data_nascita between '1980-01-01' AND '1990-12-31' AND
   giudice.sesso='f' GROUP BY te_gi_wo.nome_team;

```

### 9.13 Query 13

```

1  /*
2  si vuole sapere l'eta' media maschile e femminile di quei gruppi che
   hanno partecipato in un determinato wod
3  */
4  SELECT te_gi_wo.id_wod AS 'Id wod', te_gi_wo.nome_team AS 'Nome team',
   atleta.sesso AS 'Sesso',
5  avg(date_format(from_days(datediff(now(), atleta.data_nascita)), '%Y')+0)
   AS 'Media eta'
6  FROM atleta INNER JOIN te_gi_wo
7  ON atleta.nome_team=te_gi_wo.nome_team
8  WHERE te_gi_wo.id_wod='7m1ive554u' GROUP BY 'Nome team', 'Sesso';

```

### 9.14 Query 14

```

1  /*
2  seleziona nome del team, nome e cognome dell'atleta il quale ha il nome
   che inizia con M e termina con k e il cognome che inizia con B
3  e termina con r che ha svolto una competizione dove il nome del giudice
   inizia con E e termina con o e il cognome inizia con R e termina
4  con y. inoltre vogliamo sapere in che box e' stata svolta la
   competizione.
5  */
6  SELECT te_gi_wo.nome_team, atleta.nome AS 'Atleta: Nome', atleta.cognome
   AS 'Atleta: cognome', box.nome_box AS 'Nome box'
7  FROM atleta INNER JOIN te_gi_wo
8  ON atleta.nome_team=te_gi_wo.nome_team
9  INNER JOIN giudice
10 ON te_gi_wo.id_giudice=giudice.id_giudice
11 INNER JOIN wod
12 ON te_gi_wo.id_wod=wod.id_wod
13 INNER JOIN box
14 ON wod.nome_box=box.nome_box
15 WHERE atleta.nome LIKE 'M%k' AND atleta.cognome LIKE 'B%r' AND giudice
   .nome LIKE 'E%o' AND giudice.cognome LIKE 'R%y';

```

### 9.15 Query 15

```

1  /*
2  seleziona i nomi dei team l'id dei wod e il nome dell'esercizio dei wod
   che contengono esercizi con un determinato
3  numero di ripetizioni e di quei team che hanno uno o piu coach con una
   certificazione di livello 4
4  */
5  SELECT te_gi_wo.nome_team, wo_es.id_wod, wo_es.nome_esercizio
6  FROM wo_es INNER JOIN te_gi_wo
7  ON wo_es.id_wod=te_gi_wo.id_wod
8  INNER JOIN coach
9  ON te_gi_wo.nome_team=coach.nome_team
10 INNER JOIN co_ce
11 ON coach.e_mail = co_ce.e_mail
12 WHERE wo_es.ripetizioni=15 AND co_ce.id_livello='cf-14';

```

## 9.16 Query 16

```
1  /*
2  fornita una data di una competizione vogliamo sapere chi era presente
   come spettatore ottenendo il nome e il cognome e i nomi dei team
3  che hanno visto gareggiare
4  */
5  SELECT persona.nome, persona.cognome, te_gi_wo.nome_team
6  FROM te_gi_wo INNER JOIN bo_pe
7  ON bo_pe.data_presenza=te_gi_wo.data_esecuzione
8  INNER JOIN persona
9  ON bo_pe.codice_fiscale=persona.codice_fiscale
10 WHERE te_gi_wo.data_esecuzione='2015-08-28';
```

## 9.17 Query 17

```
1  /*
2  vogliamo sapere il nome e il cognome di tutti gli spettatori di un
   determinato team e il nome dei box in cui hanno
3  seguito il team
4  */
5  SELECT persona.nome, persona.cognome, wod.nome_box, wod.id_wod
6  FROM te_gi_wo, wod, bo_pe, persona
7  WHERE te_gi_wo.id_wod=wod.id_wod AND wod.nome_box=bo_pe.nome_box AND
   bo_pe.codice_fiscale=persona.codice_fiscale AND te_gi_wo.nome_team=
   'team10' GROUP BY nome, cognome;
```

## 9.18 Query 18

```
1  /*
2  selezionare tutte le persone che hanno fatto da spettatori con lo
   stesso nome di un giudice e di queste persone vogliamo sapere dove
   hanno fatto
3  da spettatori e in che data
4  */
5  SELECT distinct p1.nome AS 'Nome', p1.cognome AS 'Cognome', bo_pe.
   nome_box AS 'Nome box', te_gi_wo.data_esecuzione AS 'Data
   partecipazione'
6  FROM persona AS p1, giudice AS p2, bo_pe, box, wod, te_gi_wo
7  WHERE p1.nome=p2.nome AND p1.cognome!=p2.cognome AND p1.codice_fiscale=
   bo_pe.codice_fiscale AND bo_pe.nome_box=box.nome_box
8  AND wod.nome_box=box.nome_box AND wod.id_wod=te_gi_wo.id_wod;
```

## 10 Algebra Relazionale

In questa sezione si utilizzerà l'algebra relazionale per rappresentare alcune query che precedentemente sono state presentate in SQL.

### 10.1 Query 1 in Algebra relazionale

Nella prima query abbiamo utilizzato l'entità COACH e la relazione CO\_CE che sono rappresentate nello schema Entity Relationship, mentre le identifichiamo come entità fisiche nello Schema Fisico.

$$\begin{aligned} \pi_{nome,cognome,livello1,livello2,livello3,livello4}(\rho_{livello1 \leftarrow id\_livello}(CO\_CE) \\ \bowtie \rho_{livello2 \leftarrow id\_livello}(CO\_CE) \bowtie \rho_{livello3 \leftarrow id\_livello}(CO\_CE) \bowtie \rho_{livello4 \leftarrow id\_livello}(CO\_CE)) \end{aligned} \quad (1)$$

### 10.2 Query 2 in Algebra Relazionale

La seconda query, utilizza solo l'entità COACH per recuperare i dati che si vogliono ottenere. In Algebra relazionale possiamo ottenere lo stesso risultato come riportato di seguito.

$$\pi_{nome,cognome,sesto}(\sigma_{nome\_team='team01'}(COACH)) \quad (2)$$



## 11 Calcolo Relazionale

Nella sezione riguardante l'Algebra Relazionale, abbiamo visto come trasformare le prime due query SQL in algebra relazionale. Ora vogliamo esprimere sempre le prime due query SQL in Calcolo Relazionale. Di seguito sono riportate le trasformazioni delle query.

### 11.1 Query 1 in Calcolo Relazionale

In questa query utilizziamo due schemi di relazione per ottenere i dati che si ottengono con la query SQL.

$$\begin{aligned} & \{nome : n, cognome : c, id\_livello : l1, \\ & id\_livello : l2, id\_livello : l3, id\_livello : l4 | \\ & COACH(COACH(e\_mail : e, nome : n, cognome : c) \\ & \wedge CO\_CE(e\_mail : e, id\_livello : l1)) \\ & \wedge CO\_CE(e\_mail : e, id\_livello : l2)) \\ & \wedge CO\_CE(e\_mail : e, id\_livello : l3)) \\ & \wedge CO\_CE(e\_mail : e, id\_livello : l4))\} \end{aligned} \quad (3)$$

### 11.2 Query 2 in Calcolo Relazionale

In questa query utilizzeremo solo uno schema di relazione come nella query SQL.

$$\begin{aligned} & \{nome : n, cognome : c, sesso : s | \\ & COACH(nome : n, cognome : c, sesso : s) \\ & \wedge nome\_team = 'team01'\} \end{aligned} \quad (4)$$

## 12 Trigger

Per risolvere i problemi legati alle regole di vincolo si possono utilizzare i trigger. Di seguito sono riportati i listati dei trigger realizzati.

### 12.1 Trigger 1

Questo trigger si occupa di determinare se un atleta che si sta inserendo è maggiorenne e verificherà che la mail inserita sia sintatticamente corretta.

```
1 DELIMITER //
2 -- Nome: data_check_atleta
3 --
4 CREATE TRIGGER data_check_atleta BEFORE INSERT ON atleta
5 FOR EACH ROW
6 BEGIN
7     IF NEW.data_nascita < DATE('1997-01-01') THEN
8         SIGNAL SQLSTATE VALUE '45000'
9         SET MESSAGE_TEXT = '[table:persona] - 'data_nascita' data di
            nascita non valida.';
10    END IF;
11
12    IF NEW.e_mail NOT REGEXP '^[A-Z0-9._%~]+@[A-Z0-9.-]+\.[A-Z]{2,4}$'
13    THEN
14        SIGNAL SQLSTATE VALUE '45000'
15        SET MESSAGE_TEXT = '[table:persona] - 'e_mail' formato della mail
            non valido.';
16    END IF;
17 END //
```

### 12.2 Trigger 2

Come il trigger precedente ma è associato alla tabella coach.

```
1
2 DELIMITER //
3 -- Nome: data_check_coach
4 --
5 CREATE TRIGGER data_check_coach BEFORE INSERT ON coach
6 FOR EACH ROW
7 BEGIN
8     IF NEW.data_nascita < DATE('1997-01-01') THEN
9         SIGNAL SQLSTATE VALUE '45000'
10        SET MESSAGE_TEXT = '[table:persona] - 'data_nascita' data di
            nascita non valida.';
11    END IF;
12
13    IF NEW.e_mail NOT REGEXP '^[A-Z0-9._%~]+@[A-Z0-9.-]+\.[A-Z]{2,4}$'
14    THEN
15        SIGNAL SQLSTATE VALUE '45000'
16        SET MESSAGE_TEXT = '[table:persona] - 'e_mail' il formato della
            mail non valido.';
17    END IF;
18 END //
```

## 12.3 Trigger 3

Questo ultimo trigger si occupa solamente di verificare se si sta inserendo un giudice maggiorenne.

```
1 DELIMITER //
```

```
2  -- Nome: data_check_giudice
```

```
3  --
```

```
4 CREATE TRIGGER data_check_giudice BEFORE INSERT ON giudice
```

```
5 FOR EACH ROW
```

```
6 BEGIN
```

```
7     IF NEW.data_nascita < DATE( '1997-01-01' ) THEN
```

```
8         SIGNAL SQLSTATE VALUE '45000'
```

```
9         SET MESSAGE_TEXT = '[table:persona] - 'data_nascita' data di
```

```
10            nascita non valida.';
```

```
11     END IF;
```

```
12 END //
```

```
13 DELIMITER ;
```

## 13 Stored Procedure

Siamo giunti alle stored procedure, queste sono state utilizzate per caricare la nostra base di dati in maniera random così che si possano fare dei test per l'ottimizzazione della base di dati realizzata.

### 13.1 Procedura 1

Questa procedura ci serve per riempire le entità atleta,coach,team in maniera random. Tutti i dati che questa procedura genera e inserisce nella base di dati,sono stati generati tramite delle funzioni presenti nel DBMS utilizzato.

```
1 DELIMITER //
```

```
2  -- Nome: loaddb
```

```
3  -- Parametri: num_tentativi = numero di tentativi effettuati per
```

```
4             -- inserire i dati generati nelle entita' atleta,team, coach.
```

```
5 CREATE PROCEDURE loaddb(
```

```
6     IN numtentativi INTEGER
```

```
7 ) COMMENT 'carica database'
```

```
8 BEGIN
```

```
9     DECLARE dinizio DATE DEFAULT '2015-01-01';
```

```
10    DECLARE dfine DATE DEFAULT '2015-01-14';
```

```
11    DECLARE dnasinizio DATE DEFAULT '1955-01-01';
```

```
12    DECLARE dnasfine DATE DEFAULT '2015-12-31';
```

```
13    DECLARE dnascita DATE;
```

```
14    DECLARE discrizione DATE;
```

```
15    DECLARE count INT;
```

```
16    DECLARE counta INT;
```

```
17    DECLARE nteam VARCHAR(255);
```

```
18    DECLARE stato VARCHAR(255);
```

```
19    DECLARE email VARCHAR(255);
```

```
20    DECLARE nome VARCHAR(255);
```

```
21    DECLARE cognome VARCHAR(255);
```

```
22    DECLARE ases VARCHAR(255);
```

```
23    DECLARE rand_cert INTEGER;
```

```
23 SET count = 1;
24 SET counta = 1;
25
26 CREATE TABLE IF NOT EXISTS stati(st VARCHAR(255));
27 IF (SELECT count(*) FROM stati) = 0 THEN
28     INSERT INTO stati(st) VALUES (
29         'italy'),('france'),('germany'),('united kingdom'),('USA'
30     );
31 END IF;
32
33 CREATE TABLE IF NOT EXISTS sesso(s ENUM('M','F'));
34 IF (SELECT count(*) FROM sesso) = 0 THEN
35     INSERT INTO sesso(s) VALUES ('M'),('F');
36 END IF;
37
38 WHILE count <= numtentativi DO
39     crea un team
40     SELECT SUBSTR(MD5(RAND()),1,(5+RAND()*20)) INTO nteam;
41     SELECT FROM_DAYS(floor(TO_DAYS(dinizio)+rand()*(TO_DAYS(dfine)-
42     TO_DAYS(dinizio)))) INTO descrizione;
43     SELECT st FROM stati ORDER BY rand() LIMIT 1 INTO stato;
44     INSERT INTO team(nome,data_iscrizione,stato)VALUES(nteam,
45     descrizione,stato);
46
47     SELECT CONCAT(SUBSTR(MD5(RAND()),1,(5+RAND()*20)),'@',SUBSTR(MD5(
48     RAND()),1,(5+RAND()*20)),'it') INTO email;
49     SELECT SUBSTR(MD5(RAND()),1,(5+RAND()*20)) INTO nome;
50     SELECT SUBSTR(MD5(RAND()),1,(5+RAND()*20)) INTO cognome;
51     SELECT FROM_DAYS(floor(TO_DAYS(dnasinizio)+rand()*(TO_DAYS(dnasfine)
52     -TO_DAYS(dnasinizio)))) INTO dnascita;
53     SELECT s FROM sesso ORDER BY rand() LIMIT 1 INTO ases;
54     INSERT INTO coach(e_mail,nome_team,nome,cognome,data_nascita,sesso)
55     VALUES (
56         email,nteam,nome,cognome,dnascita,ases
57     );
58     SELECT FLOOR(1 + RAND()*4) INTO rand_cert;
59     IF rand_cert = 1 THEN
60         INSERT INTO co_ce (e_mail, id_livello) VALUES (email, "cf-11");
61     END IF;
62
63     IF rand_cert = 2 THEN
64         INSERT INTO co_ce (e_mail, id_livello) VALUES (email, "cf-11");
65         INSERT INTO co_ce (e_mail, id_livello) VALUES (email, "cf-12");
66     END IF;
67
68     IF rand_cert = 3 THEN
69         INSERT INTO co_ce (e_mail, id_livello) VALUES (email, "cf-11");
70         INSERT INTO co_ce (e_mail, id_livello) VALUES (email, "cf-12");
71         INSERT INTO co_ce (e_mail, id_livello) VALUES (email, "cf-13");
72     END IF;
73
74     IF rand_cert = 4 THEN
75         INSERT INTO co_ce (e_mail, id_livello) VALUES (email, "cf-11");
76         INSERT INTO co_ce (e_mail, id_livello) VALUES (email, "cf-12");
77         INSERT INTO co_ce (e_mail, id_livello) VALUES (email, "cf-13");
78         INSERT INTO co_ce (e_mail, id_livello) VALUES (email, "cf-14");
79     END IF;
80
81     WHILE counta <= 2 DO
82         SELECT CONCAT(SUBSTR(MD5(RAND()),1,(5+RAND()*20)),'@',SUBSTR(MD5(
83         RAND()),1,(5+RAND()*20)),'it') INTO email;
84         SELECT SUBSTR(MD5(RAND()),1,(5+RAND()*20)) INTO nome;
85         SELECT SUBSTR(MD5(RAND()),1,(5+RAND()*20)) INTO cognome;
```

```
78      SELECT FROM DAYS(floor(TO_DAYS(dnasinizio)+rand()*(TO_DAYS(
      dnasfine)-TO_DAYS(dnasinizio)))) INTO dnascita;
79      SELECT s FROM sesso ORDER BY rand() LIMIT 1 INTO ases;
80      INSERT INTO atleta(e_mail,nome_team,nome,cognome,data_nascita,
      sesso) VALUES (
81          email,nteam,nome,cognome,dnascita,ases
82      );
83      SET counta = counta + 1;
84      END WHILE;
85      SET count = count +1;
86      SET counta = 1;
87      END WHILE;
88      DROP TABLE stati;
89      DROP TABLE sesso;
90  END //
91  DELIMITER ;
```

## 13.2 Procedura 2

La procedura seguente ci permette di riempire in maniera random la relazione bo\_pe, dove ogni tupla rappresenta uno spettatore di un box.

```
1  DELIMITER //
2  -- Nome: load_spettatori
3  -- Parametri: num_tentativi = numero di tentativi effettuati per
   inserire una tupla che identifica uno spettatore di un box
4  CREATE PROCEDURE load_bo_pe(
5      IN num_tentativi INTEGER
6  ) COMMENT 'riempie la relazione bo_pe in modo random'
7  BEGIN
8      -- dichiara una variabile per il nome del box
9      DECLARE boxn VARCHAR(255);
10     -- dichiara una variabile per il codice fiscale dello spettatore
11     DECLARE cf VARCHAR(255);
12     DECLARE data_p DATE;
13     -- dichiara una variabile che fara da contatore e questa viene settata
       a 1
14     DECLARE count INT;
15     SET count = 1;
16     -- il ciclo e' utilizzato per riempire la relazione bo_pe in modo
       random
17     -- inserendo un certo numero di tuple che rappresentano gli spettatori
       di un box
18     WHILE count <= num_tentativi DO
19         -- seleziona un nome radom di un box e lo assegna a boxn
20         SELECT nome_box
21         from box
22         ORDER BY rand() LIMIT 1
23         INTO boxn;
24         -- seleziona un cdodice fiscale random di una persona e lo assegna a
           cf
25         SELECT codice_fiscale
26         FROM persona
27         ORDER BY rand() LIMIT 1
28         INTO cf;
29         -- seleziona una data random di una competizione e l'assegna a
           data_p
30         SELECT data_esecuzione
31         FROM te_gi_wo
```

```
32 ORDER BY rand() LIMIT 1
33 INTO data_p;
34 — inserisce i dati recuperati nella relazione bo_pe
35 INSERT IGNORE INTO bo_pe (nome_box, codice_fiscale, data_presenza)
VALUES (boxn, cf, data_p);
36 — incrementa la variabile contatore
37 SET count = count + 1;
38 END WHILE;
39 END //
40 DELIMITER ;
```

### 13.3 Procedura 3

Questa procedura ci permette di caricare delle tuple nella relazione `te_gi_wo` la quale rappresenta il cuore della base di dati.

```
1 DELIMITER //
2 — Nome: load_esecuzione
3 — Parametri: num_tentativi = numero di tentativi effettuati per
   inserire una tupla che identifica l'esecuzione di un wod
4 CREATE PROCEDURE load_esecuzione(
5   IN num_tentativi INTEGER
6 )COMMENT 'riempie la relazione te_gi_wo in modo random'
7 BEGIN
8   — variabili utilizzate per memorizzare i dati recuperati in modo
   random
9
10  — usata per il nome del team
11  DECLARE nome_t VARCHAR(255);
12  — usata per l'id del giudice
13  DECLARE id_g VARCHAR(255);
14  — usata per l'id del wod
15  DECLARE id_w VARCHAR(255);
16  — usata per la data di pubblicazione del wod
17  DECLARE d_pub DATE;
18  — usata per la data di termine del wod
19  DECLARE d_term DATE;
20  — usata per memorizzare una data ricavata da un'intervallo di date
21  DECLARE d_ins DATE;
22  — usata per memorizzare il punteggio ricavato random
23  DECLARE score INT;
24  — variabile contatore
25  DECLARE count INT;
26  — usata per memorizzare il punteggio minimo
27  DECLARE min_score INT;
28  — usata per memorizzare il punteggio massimo
29  DECLARE max_score INT;
30  — imposta il punteggio minimo assegnabile
31  SET min_score = 0;
32  — imposta il punteggio massimo assegnabile
33  SET max_score = 100;
34  — inizializza il contatore
35  SET count = 1;
36
37  — il ciclo viene effettuato fintanto che il valore di count non e'
   uguale a num_tentativi, mentre nel corpo del while si
38  — recuperano dei dati in modo random, quali: nome, id_giudice, id_wod
   , data_pubblicazione, data_termine e viene generata
```

```
39  -- una data random compresa tra data_publicazione e data_termine ,
    questa data identifica il giorno in cui il team esegue il
40  -- wod, infine viene inserita una tupla nella relazione te_gi_wo con i
    dati recuperati e la data generata.
41  WHILE count <= num_tentativi DO
42
43      SELECT nome INTO nome_t from team ORDER BY RAND() LIMIT 1;
44      SELECT id_giudice INTO id_g FROM giudice ORDER BY RAND() LIMIT 1;
45      SELECT id_wod INTO id_w FROM wod ORDER BY RAND() LIMIT 1;
46      SELECT data_publicazione INTO d_pub FROM wod WHERE id_wod = id_w;
47      SELECT data_termine INTO d_term FROM wod WHERE id_wod = id_w;
48
49      SELECT FROM_DAYS(floor (TO_DAYS(d_pub)+rand()*(TO_DAYS(d_term)-
    TO_DAYS(d_pub)))) INTO d_ins;
50      SELECT FLOOR(min_score + rand() * (max_score - min_score)) INTO
    score;
51
52      INSERT IGNORE INTO te_gi_wo(id_giudice, id_wod, data_esecuzione,
    nome_team, punteggio) VALUES (id_g, id_w, d_ins, nome_t, score);
53
54      SET count = count + 1;
55
56  END WHILE;
57 END //
58 DELIMITER ;
```

## 13.4 Procedura 4

Per inserire le tuple che rappresentano il possesso di una certa certificazione nella relazione `co_ce` utilizziamo la stored procedure seguente.

```
1  DELIMITER //
2  -- Nome: load_co_ce
3  -- Parametri: nessun parametro
4  CREATE PROCEDURE load_co_ce(
5  )COMMENT 'riempie la relazione co_ce in modo random'
6  BEGIN
7  -- variabile per memorizzare la mail del coach
8  DECLARE mail VARCHAR(255);
9  -- variabile per contenere un valore numerico intero
10 DECLARE rand_cert INT;
11 -- variabile booleana per verificare lo stato di una condizione
12 DECLARE done BOOLEAN DEFAULT 0;
13 -- dichiarazione di un cursore sulla tabella coach
14 DECLARE tmpcoach CURSOR FOR SELECT e_mail FROM coach;
15 -- dichiara un handler che intercetta lo stato di errore 02000 e setta
16 -- la variabile booleana done a 1
17 DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done=1;
18 -- apre il cursore creato in precedenza
19 OPEN tmpcoach;
20 -- qui viene ripetuto il codice tra repeat e end repeat fintanto che l'
    handler
21 -- non intercetta lo stato di errore o vero non ci sono piu record da
    scorrere
22 -- nella tabella coach tramite il cursore.
23 -- per ogni coach recuperato gli si assegna random un certo numero di
    certificazioni.
24 REPEAT
25  FETCH tmpcoach INTO mail;
```

```
26
27 SELECT FLOOR(1 + RAND()*4) INTO rand_cert;
28 IF rand_cert = 1 THEN
29     INSERT INTO co_ce (e_mail, id_livello) VALUES (mail, "cf-11");
30 END IF;
31
32 IF rand_cert = 2 THEN
33     INSERT INTO co_ce (e_mail, id_livello) VALUES (mail, "cf-11");
34     INSERT INTO co_ce (e_mail, id_livello) VALUES (mail, "cf-12");
35 END IF;
36 IF rand_cert = 3 THEN
37     INSERT INTO co_ce (e_mail, id_livello) VALUES (mail, "cf-11");
38     INSERT INTO co_ce (e_mail, id_livello) VALUES (mail, "cf-12");
39     INSERT INTO co_ce (e_mail, id_livello) VALUES (mail, "cf-13");
40 END IF;
41 IF rand_cert = 4 THEN
42     INSERT INTO co_ce (e_mail, id_livello) VALUES (mail, "cf-11");
43     INSERT INTO co_ce (e_mail, id_livello) VALUES (mail, "cf-12");
44     INSERT INTO co_ce (e_mail, id_livello) VALUES (mail, "cf-13");
45     INSERT INTO co_ce (e_mail, id_livello) VALUES (mail, "cf-14");
46 END IF;
47
48 UNTIL done END REPEAT;
49 — chiude il cursore
50 CLOSE tmpcoach;
51 END //
52 DELIMITER ;
```



## 14 Ottimizzazioni

In questa sezione, verranno analizzate delle query che risultano molto lente e si cercherà di ottimizzare queste query al fine di renderle più performanti.

### 14.1 ottimizzazione query 4

Questa query impiega: 12 rows in set (10,10 sec) Dopo aver applicato lo STRAIGHT JOIN che obbliga l'ottimizzatore a valutare le tabelle presenti nella from con la sequenza specificata e non come lui crede più opportuno otteniamo un miglioramento in termini di tempo pari a: 12 rows in set (2,89 sec). Di seguito riportiamo la query prima e dopo l'ottimizzazione e le explain di tali query.

#### 14.1.1 Query non ottimizzata

```
1 SELECT num_state.stato,num_state.'team per stato',avg_age.'eta' media '
2 FROM (select stato,count(stato) as 'team per stato '
3      from team group by stato) AS num_state
4 INNER JOIN
5 (SELECT stato , avg(date_format(from_days(datediff(now()),atleta.
6      data_nascita)),'%Y')+0) AS 'eta' media '
7 FROM team,atleta
8 WHERE atleta.nome_team=team.nome group by team.stato) AS avg_age
9 ON num_state.stato=avg_age.stato;
```

#### 14.1.2 Query ottimizzata

```
1 SELECT num_state.stato,num_state.'team per stato',avg_age.'eta' media '
2 FROM (select stato,count(stato) as 'team per stato '
3      from team group by stato) AS num_state
4 INNER JOIN
5 (SELECT /*! STRAIGHT_JOIN */ stato , avg(date_format(from_days(
6      datediff(now()),atleta.data_nascita)),'%Y')+0) AS 'eta' media '
7 FROM atleta,team
8 WHERE atleta.nome_team=team.nome group by team.stato) AS avg_age
9 ON num_state.stato=avg_age.stato;
```

#### 14.1.3 Explain query non ottimizzata

```
1 ***** 1. row *****
2      id: 1
3      select_type: PRIMARY
4      table: <derived2>
5      type: ALL
6 possible_keys: NULL
7      key: NULL
8      key_len: NULL
9      ref: NULL
10     rows: 302878
11     Extra: NULL
12 ***** 2. row *****
13      id: 1
```

```

14   select_type: PRIMARY
15     table: <derived3>
16     type: ref
17 possible_keys: <auto_key0>
18     key: <auto_key0>
19     key_len: 767
20     ref: num_state.stato
21     rows: 10
22     Extra: NULL
23 ***** 3. row *****
24     id: 3
25     select_type: DERIVED
26     table: team
27     type: ALL
28 possible_keys: PRIMARY
29     key: NULL
30     key_len: NULL
31     ref: NULL
32     rows: 302878
33     Extra: Using temporary; Using filesort
34 ***** 4. row *****
35     id: 3
36     select_type: DERIVED
37     table: atleta
38     type: ref
39 possible_keys: nome_team
40     key: nome_team
41     key_len: 767
42     ref: crossfit2.team.nome
43     rows: 1
44     Extra: NULL
45 ***** 5. row *****
46     id: 2
47     select_type: DERIVED
48     table: team
49     type: ALL
50 possible_keys: NULL
51     key: NULL
52     key_len: NULL
53     ref: NULL
54     rows: 302878
55     Extra: Using temporary; Using filesort
56 5 rows in set (0,01 sec)

```

#### 14.1.4 Explain query ottimizzata

```

1 ***** 1. row *****
2     id: 1
3     select_type: PRIMARY
4     table: <derived2>
5     type: ALL
6 possible_keys: NULL
7     key: NULL
8     key_len: NULL
9     ref: NULL
10    rows: 302878
11    Extra: NULL
12 ***** 2. row *****
13     id: 1
14     select_type: PRIMARY
15     table: <derived3>

```

```

16         type: ref
17 possible_keys: <auto_key0>
18         key: <auto_key0>
19         key_len: 767
20         ref: num_state.stato
21         rows: 10
22         Extra: NULL
23 ***** 3. row *****
24         id: 3
25         select_type: DERIVED
26         table: atleta
27         type: ALL
28 possible_keys: nome_team
29         key: NULL
30         key_len: NULL
31         ref: NULL
32         rows: 594675
33         Extra: Using temporary; Using filesort
34 ***** 4. row *****
35         id: 3
36         select_type: DERIVED
37         table: team
38         type: eq_ref
39 possible_keys: PRIMARY
40         key: PRIMARY
41         key_len: 767
42         ref: crossfit2.atleta.nome_team
43         rows: 1
44         Extra: NULL
45 ***** 5. row *****
46         id: 2
47         select_type: DERIVED
48         table: team
49         type: ALL
50 possible_keys: NULL
51         key: NULL
52         key_len: NULL
53         ref: NULL
54         rows: 302878
55         Extra: Using temporary; Using filesort
56 5 rows in set (0,01 sec)

```

## 14.2 ottimizzazione query 7

Questa query impiega: 12764 rows in set (3,58 sec) Anche in questa query viene utilizzato lo STRAIGHT JOIN. In questo caso si è ottenuto un miglioramento nel tempo di risposta pari a: 12764 rows in set (0,23 sec).

### 14.2.1 Query non ottimizzata

```

1 SELECT te_gi_wo.nome_team,te_gi_wo.data_esecuzione,te_gi_wo.id_wod,
   te_gi_wo.punteggio,giudice.nome,giudice.cognome,team.stato
2 FROM te_gi_wo INNER JOIN team
3 ON te_gi_wo.nome_team = team.nome
4 INNER JOIN giudice
5 ON te_gi_wo.id_giudice = giudice.id_giudice
6 WHERE team.stato = 'italy' AND te_gi_wo.id_wod='2rfupf2bhl' ORDER BY
   te_gi_wo.punteggio DESC;

```

### 14.2.2 Query ottimizzata

```

1 SELECT /*! STRAIGHT JOIN */ te_gi_wo.nome_team,te_gi_wo.
   data_esecuzione,te_gi_wo.id_wod,te_gi_wo.punteggio,giudice.nome,
   giudice.cognome,team.stato
2 FROM giudice,te_gi_wo,team
3 WHERE te_gi_wo.nome_team = team.nome AND te_gi_wo.id_giudice = giudice.
   id_giudice AND team.stato = 'italy' AND te_gi_wo.id_wod='2rfupf2bhl'
   ORDER BY te_gi_wo.punteggio DESC;

```

### 14.2.3 Explain query non ottimizzata

```

1 ***** 1. row *****
2      id: 1
3      select_type: SIMPLE
4      table: te_gi_wo
5      type: ref
6      possible_keys: PRIMARY,id_wod
7      key: id_wod
8      key_len: 767
9      ref: const
10     rows: 120676
11     Extra: Using index condition; Using where; Using filesort
12 ***** 2. row *****
13      id: 1
14      select_type: SIMPLE
15      table: team
16      type: eq_ref
17      possible_keys: PRIMARY
18      key: PRIMARY
19      key_len: 767
20      ref: crossfit2.te_gi_wo.nome_team
21      rows: 1
22      Extra: Using where
23 ***** 3. row *****
24      id: 1
25      select_type: SIMPLE
26      table: giudice
27      type: eq_ref
28      possible_keys: PRIMARY
29      key: PRIMARY
30      key_len: 767
31      ref: crossfit2.te_gi_wo.id_giudice
32      rows: 1
33      Extra: NULL
34 3 rows in set (0,01 sec)

```

### 14.2.4 Explain query ottimizzata

```

1 ***** 1. row *****
2      id: 1
3      select_type: SIMPLE
4      table: giudice
5      type: ALL
6      possible_keys: PRIMARY
7      key: NULL
8      key_len: NULL
9      ref: NULL
10     rows: 24
11     Extra: Using temporary; Using filesort

```

```

12 ***** 2. row *****
13      id: 1
14      select_type: SIMPLE
15      table: te_gi_wo
16      type: ref
17 possible_keys: PRIMARY, id_wod
18      key: PRIMARY
19      key_len: 1534
20      ref: crossfit2.giudice.id_giudice,const
21      rows: 1188
22      Extra: Using where
23 ***** 3. row *****
24      id: 1
25      select_type: SIMPLE
26      table: team
27      type: eq_ref
28 possible_keys: PRIMARY
29      key: PRIMARY
30      key_len: 767
31      ref: crossfit2.te_gi_wo.nome_team
32      rows: 1
33      Extra: Using where
34 3 rows in set (0,00 sec)

```

### 14.3 ottimizzazione query 11

Questa query impiega: 28914 rows in set (0,10 sec). Per questa query abbiamo utilizzato un indice sull'attributo nome dell'entità giudice per ottenere un ordinamento. La query così ottimizzata ha ottenuto un miglioramento pari a: 28914 rows in set (0,05 sec)

```

1 SELECT te_gi_wo.nome_team, te_gi_wo.data_esecuzione, te_gi_wo.punteggio,
   wod.nome_box
2 FROM giudice INNER JOIN te_gi_wo
3 ON giudice.id_giudice=te_gi_wo.id_giudice
4 INNER JOIN wod
5 ON te_gi_wo.id_wod=wod.id_wod
6 WHERE giudice.nome='Jutta' AND giudice.cognome='Dezell';

```

#### 14.3.1 Explain query senza l'utilizzo dell'indice

```

1 ***** 1. row *****
2      id: 1
3      select_type: SIMPLE
4      table: wod
5      type: index
6 possible_keys: PRIMARY
7      key: nome_box
8      key_len: 767
9      ref: NULL
10     rows: 11
11     Extra: Using index
12 ***** 2. row *****
13      id: 1
14      select_type: SIMPLE
15      table: giudice
16      type: ALL

```

```
17 possible_keys: PRIMARY
18     key: NULL
19     key_len: NULL
20     ref: NULL
21     rows: 24
22     Extra: Using where; Using join buffer (Block Nested Loop)
23 ***** 3. row *****
24     id: 1
25     select_type: SIMPLE
26     table: te_gi_wo
27     type: ref
28 possible_keys: PRIMARY,id_wod
29     key: PRIMARY
30     key_len: 1534
31     ref: crossfit2.giudice.id_giudice,crossfit2.wod.id_wod
32     rows: 1188
33     Extra: NULL
34 3 rows in set (0,00 sec)
```

### 14.3.2 Explain query con l'utilizzo dell'indice

```
1 ***** 1. row *****
2     id: 1
3     select_type: SIMPLE
4     table: giudice
5     type: ALL
6 possible_keys: PRIMARY
7     key: NULL
8     key_len: NULL
9     ref: NULL
10    rows: 24
11    Extra: Using temporary; Using filesort
12 ***** 2. row *****
13     id: 1
14     select_type: SIMPLE
15     table: te_gi_wo
16     type: ref
17 possible_keys: PRIMARY,id_wod
18     key: PRIMARY
19     key_len: 1534
20     ref: crossfit2.giudice.id_giudice,const
21     rows: 1188
22     Extra: Using where
23 ***** 3. row *****
24     id: 1
25     select_type: SIMPLE
26     table: team
27     type: eq_ref
28 possible_keys: PRIMARY
29     key: PRIMARY
30     key_len: 767
31     ref: crossfit2.te_gi_wo.nome_team
32     rows: 1
33     Extra: Using where
34 3 rows in set (0,00 sec)
```