

Federico Ranaldi

Progetto di Basi di Dati

Περιοδονίκης

Περιοδονίκης

Il presente progetto consiste nella realizzazione di una base di dati che contenga una serie di informazioni utili alla messa in opera di una collaborazione tra diversi centri sportivi.

L'idea è quella di fornire ad alcuni centri sportivi, che dal punto di vista gestionale sono indipendenti e autonomi, un database condiviso che contenga dati riguardanti principalmente i loro iscritti e le attività che questi praticano.

Il nome del progetto ("Περιοδονίκης" in greco antico significa <<vincitore di tutti gli sport>>) deriva dall'appellativo che veniva rivolto al vincitore di tutte le prove dei giochi panellenici e l'intenzione che ci si pone nel realizzarlo è quella di promuovere il più possibile la pratica di diversi sport in diversi centri.

Come ben sappiamo dal punto di vista formale se si vuole praticare più di uno sport in diversi centri sportivi emergono tante questioni che potrebbero risultare problematiche e rendere piuttosto scomodo il perseguimento di questo obiettivo. Ad esempio supponiamo di voler praticare più di uno sport diverso in centri sportivi differenti senza doversi iscrivere ad ognuno di questi. Se ce lo permettono dovremmo perdere tempo a dare ogni volta i dati personali e quindi compilare numerose scartoffie per poi dover consegnare una copia del certificato medico ad ogni centro sportivo.

Effettivamente, chiunque intimorito da questi aspetti burocratici finirebbe per abbandonare un simile proposito.

Il dover fornire le proprie generalità così come il certificato medico sono aspetti noiosi ma comunque importanti perché i centri sportivi devono sia poter gestire gli iscritti che avere a cuore la loro salute.

Un altro aspetto del quale il progetto si interessa è quello della salute.

Infatti un eventuale iscritto vorrebbe accedere, più facilmente di quanto spesso accade, alla sanità.

Un iscritto ad uno dei centri sportivi che aderiscono al progetto deve poter rivolgersi a qualsiasi specialista senza doverlo cercare ma essendo interfacciato con esso per mezzo del proprio centro sportivo.

Accade che un centro sportivo possa essere in convenzione con uno o più centri medici/sanitari. Vogliamo che un qualsiasi centro sportivo in Περιοδονίκης (si ricorda che "Periodonikes" è il nome del progetto) possa diventare automaticamente convenzionato con un centro con il quale è già convenzionato un altro centro sportivo che ha aderito al progetto. Vogliamo inoltre che vengano registrate tutte le visite fatte da un iscritto e ciò potrebbe rivelarsi molto utile. Quindi un iscritto potrebbe accedere in lettura a parte del database che si vuole creare individuando le informazioni sul medico o sul tipo di specialista che cerca.

Riallacciandoci con l'ultimo argomento trattato vogliamo che ciò che accade per i centri medici/sanitari convenzionati e i centri sportivi aderenti al progetto, accada anche per i centri di

donazione del sangue con l'intento di promuovere la campagna di donazione ma anche per facilitare gli iscritti ad interagire con un eventuale centro di donazione. Si terrebbe conto delle donazioni fatte dagli iscritti che potrebbero essere premiati in diversi modi (ad esempio con lezioni private di uno sport scontate o gratuite oppure con sconti presso gli studi medico/sanitari).

Infine un utile espediente per la reputazione dei centri sportivi potrebbe essere quello di registrare recensioni da utenti dei vari centri sportivi e visualizzare eventuali pregi o difetti per far sì che gli aderenti al progetto possano confrontarsi. Ad esempio un centro sportivo che ha ricevuto recensioni negative sullo staff potrebbe chiedere consigli ad uno che invece ha suscitato pareri positivi.

Progettazione concettuale

In questa sezione viene fatta una panoramica su come potrebbe esser fatta la base di dati. Verranno descritte delle possibili entità che nel corso della progettazione potrebbero cambiare leggermente sia nel contenuto che nel nome.

Dunque una delle entità principali è "Centro Sportivo" e contiene per ogni record un centro sportivo che ha deciso di aderire al progetto.

Importante è capire come implementare il fatto che un iscritto che pratica più attività in diversi centri sportivi possa essere riconosciuto all'interno della base di dati.

In questa ottica avremo concettualmente due entità distinte. Una si chiamerà "Iscritto" ed ogni suo record conterrà le generalità di un iscritto e l'altra si chiamerà "Utente" e conterrà un record per ogni iscritto che pratica uno sport in un determinato centro sportivo ad un determinato livello (professionistico o dilettantistico). Ad esempio se un iscritto pratica n sport in 1 centro sportivo ad un determinato livello avremo n record. Se un iscritto pratica 1 sport ad un determinato livello in n centri sportivi avremo n record. Se un iscritto pratica n sport ad un certo livello in m centri sportivi avremo $n \cdot m$ record.

Dopodiché vogliamo catalogare gli sport praticati in tutti i centri sportivi e lo facciamo con l'entità "Sport" che tramite la relazione "Praticato" con "Centro Sportivo" ci permette di capire quali sport a quale livello sono praticati in quale centro sportivo. Si sottolinea che uno sport può essere praticato anche in più di un centro sportivo a più livelli.

È anche importante conoscere quale sia il personale tecnico che si occupa di insegnare o seguire gli utenti in una determinata attività. Analogamente ci potrebbe servire conoscere i componenti dello staff di gestione di un centro sportivo.

Abbiamo quindi bisogno di due entità che sono in relazione con "Centro Sportivo" che sono "Personale Amministrativo" e "Personale Tecnico". Quest'ultima è messa in relazione anche con "Sport".

Potrebbe essere utile come detto precedentemente raccogliere recensioni dagli iscritti su vari aspetti dei centri sportivi e lo

facciamo tramite "Recensione".Ad esempio dopo aver rilevato una serie di recensioni negative sullo staff di gestione di un centro sportivo e positive invece per quello di un altro potremo individuare i componenti del personale amministrativo di questi due e fare in modo che si possano riunire e confrontarsi sull'operato.

Veniamo alla parte riguardante la salute dell'iscritto che coinvolge l'entità "Centro Sanitario" e "Specialista".Quest'ultima contiene gli specialisti che lavorano in un determinato studio. L'entità "Visita" contiene tutte le visite fatte da un iscritto presso uno specialista che opera in uno dei centri medico/sanitari con cui un centro sportivo che ha aderito al progetto è convenzionato.

Per interfacciare l'iscritto con i centri di donazione del sangue convenzionati, abbiamo bisogno dell'entità "Centro Donazione" e l'entità "Donazione" che contiene tutte le donazioni effettuate da un iscritto.

Per quel che riguarda la gestione dei certificati medici immaginiamo che un'applicazione esterna possa elaborarne il contenuto.

Ad esempio verosimilmente potrebbe esserci un portale online al quale accedere tramite codice fiscale dell'utente e un secondo codice, che funge da password per ottenere il file contenente il certificato medico.

All'interno dell'entità "Iscritto" avremo quindi un attributo 'Codice-Certificato_Medico' che serve appunto per ottenere il certificato dell'utente.Se il valore attribuito a questo campo è None(o null) allora significa che o il certificato è scaduto oppure che non è presente.Quindi si dovrà provvedere ad avvertire l'utente affinché faccia/rinnovi il certificato e magari ad indirizzarlo verso un medico di uno studio convenzionato. La gestione della veridicità o della scadenza dei certificati non rientra tra i requisiti di questo progetto e va gestita esternamente.A noi interessa che ogni Centro Sportivo aderente a Περιοδονίκης possa accedere al certificato dell'iscritto/utente senza doverglielo chiedere.

Glossario dei termini

Nel corso della presentazione alcune asserzioni potrebbero generare ambiguità e potrebbe non essere chiaro il significato di alcune entità o relazioni. Il seguente glossario specifica a cosa ci si riferisce quando si utilizzano certi termini.

TERMINE	DESCRIZIONE	SINONIMI	COLLEGAMENTI
Iscritto	Identifica univocamente una persona che frequenta uno o più centri sportivi	Persona, Individuo, Frequentatore...	Utente, Visita, Centro Sportivo, Recensione, Donazione, Utente
Utente	Identifica un iscritto che pratica un determinato sport in un determinato centro sportivo". N.B.:un iscritto se pratica più sport in diversi in diversi centri sportivi può corrispondere a più utenti	Praticante di uno sport, Sportivo...	Iscritto, Sport
Sport	Nome di un'attività che può essere praticata in diversi centri sportivi ad un determinato livello	Attività, Disciplina ...	Personale Tecnico, Centro Sportivo, Utente
Personale Tecnico	Insieme di esperti che si occupa di seguire gli utenti in uno o più sport in uno o più centri sportivi N.B.:un membro del personale tecnico può lavorare in più di un centro sportivo che aderisce a Periodonikés	Istruttori, Coach, Maestri, Componenti dello staff tecnico...	Centro Sportivo, Sport

TERMINI	DESCRIZIONE	SINONIMI	COLLEGAMENTI
Personale Amministrativo	Insieme di impiegati o dirigenti che si occupano di gestire dal punto di vista logistico un centro sportivo N.B:un membro del personale amministrativo può lavorare in uno solo de centri sportivi che aderiscono a Periodonikés	Impiegati, Componenti dello staff amministrativo... (include segretari etc...)	Centro Sportivo
Centro Sanitario	Luogo in cui si possono effettuare visite o terapie volte a salvaguardare la salute fisica e mentale.Se ad opervarvi sono solo medici si parla di "Studio Medico"	Centro Medico/Sanitario, Studio Medico(improprio)..	Centro Sportivo, Specialista
Specialista	Persona che opera come esperto in un Centro Sanitario.Include medici, fisioterapisti,... quindi che esercitano nel campo sanitario.	Esperto, Medico(improprio), Dottore(improprio)...	Visita, Centro Sanitario
Visita	Incontro fissato di una persona tra gli iscritti con uno specialista di un centro sanitario.Può essere una visita medica,una seduta da fisioterapista, analisi...	Appuntamento, Incontro, Esame(improprio),Controllo(improprio)..	Iscritto, Specialista
Recensione	Riga contenente un'opinione espressa da un iscritto su un determinato aspetto	Opinione,Parere, Commento(improprio), Critica(improprio)..	Iscritto

	di un centro sportivo sotto forma di voto(valore compreso in un range)+descrizione.		
--	---	--	--

Vincoli

È importante chiarire cosa rappresentano effettivamente i componenti della base di dati in modo tale che non ci siano dubbi su cosa questi possono contenere.

Sono elencati una serie di vincoli:

- Si impone che un iscritto possa essere iscritto solamente ad uno dei Centri Sportivi che aderiscono al progetto.
- Un utente può praticare un solo sport in un solo centro sportivo.
- Non sono contemplati sport che non vengono praticati in nessuno dei centri sportivi.
- Un membro del personale tecnico può essere competente in più di uno sport.
- Un membro del personale tecnico può lavorare in più di un centro sportivo aderente al progetto.
- Un membro del personale amministrativo può lavorare in un solo centro sportivo aderente al progetto.
- Non sono contemplati centri sanitari non convenzionati con nessuno dei centri sportivi.
- Un centro sanitario può essere convenzionato con più di uno dei centri sportivi aderenti al progetto.
- Non sono contemplati centri donazione non convenzionati con nessuno dei centri sportivi.
- Un centro donazione può essere convenzionato con più di uno dei centri sportivi aderenti al progetto.
- Uno specialista può lavorare in un solo centro sanitario.
- Una recensione può esser fatta su un solo centro sportivo e solo da un utente che lo frequenta.
- Un utente che frequenta un centro sportivo può effettuare una recensione solo una volta ogni 5 mesi circa.
- Un iscritto può effettuare una donazione solo ogni 3 mesi circa. Ovviamente non possiamo controllare la vita dell'iscritto. Il database deve impedire insert sulla tabella "Donazione" che non rispettino quanto scritto sopra.

Regole di derivazione

- Il numero di iscritti per ogni centro sportivo si può ricavare dall'entità "Iscritto".
- Gli sport praticati da un iscritto in un determinato centro sportivo si ottengono dall'entità "Utenti".
- La media delle opinioni su un centro sportivo si ottiene operando matematicamente sull'insieme delle recensioni fatte su un centro

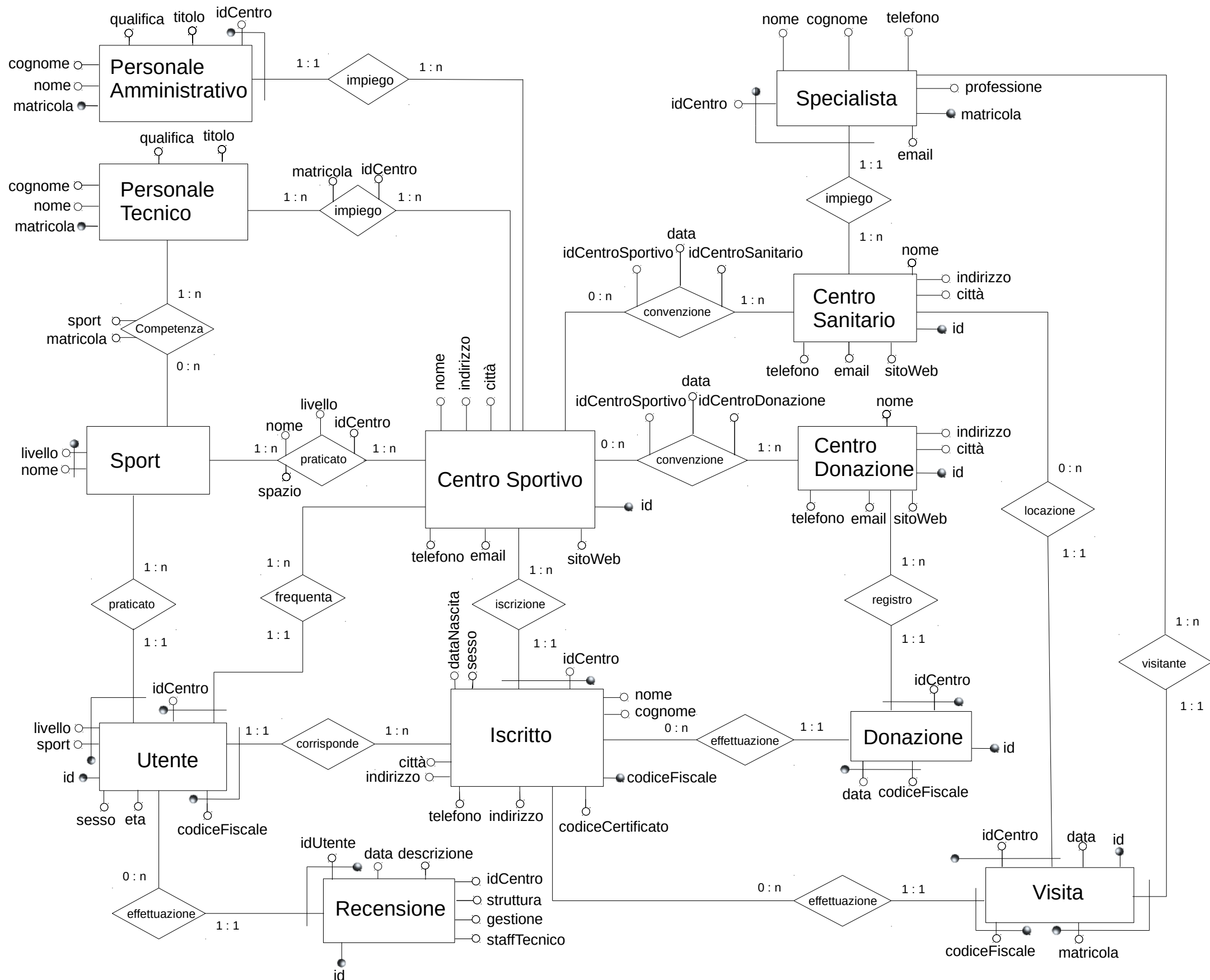
sportivo.

-Si possono ottenere i contatti di una serie di studi medici per ottenere visite presso specialisti esperti in un determinato campo selezionando operando sulle entità "Centri Sanitari" e "Specialista".

-Il numero di frequentatori(che è diverso da iscritti) di un centro sportivo si deriva dall'entità "Utente" come il numero di record aventi diverso valore dell'attributo codiceFiscale e stesso di centroSportivo.

-L'età di un utente si ricava dalla data di nascita dell'iscritto che lo rappresenta.

Schema ER-logico



Relazioni

Nella progettazione concettuale è spiegato a grandi linee ciò che le entità raffigurate nello schema ER-logico rappresentano.

In questa sezione sono descritte in maniera approssimativa le relazioni che intercorrono tra queste. Capire il significato delle relazioni è molto importante essendo queste un aspetto cruciale nel passaggio allo Schema ER-fisico.

In forma tabellare è spiegato il significato di una relazione e delle sue cardinalità. Avremo quindi il nome della relazione, le 2 entità coinvolte, poi il significato della relazione ed infine la spiegazione delle cardinalità il cui valore è visualizzabile nello schema. Si nota che alcune relazioni hanno lo stesso nome seppur coinvolgano entità differenti. Effettivamente ciò è sbagliato, ma nello schema ER-fisico a partire dal quale si effettuerà l'implementazione il problema non ci sarà più.

Nello schema ER-logico, che ha un significato più concettuale che pratico si è preferito lasciare che più entità potessero avere lo stesso nome.

*per quanto riguarda le cardinalità sono spiegate quelle meno scontate

Relazione	Entità_1	Entità_2	Significato	Cardinalità
iscrizione	Centro Sportivo	Iscritto	Quale persona è iscritta a quale centro sportivo	Una o più persone possono iscriversi a un centro sportivo. Una persona può iscriversi a un solo centro sportivo
praticato	Centro Sportivo	Sport	Quale sport è praticato in quale centro sportivo	In un centro sportivo deve praticarsi almeno uno sport che può essere praticato in più centri sportivi
corrisponde	Iscritto	Utente	A quale iscritto corrispondono uno o più utenti	Per un iscritto deve esserci almeno un utente. Un utente può rappresentare un solo iscritto
pratica	Utente	Sport	Quale sport praticano uno o più utenti	Uno sport deve avere almeno un utente per essere praticato.
impiego	Centro	Personale	Quale membro	Un membro del

	Sportivo	Tecnico	del personale tecnico ha un impiego in uno o più dei centri sportivi	personale tecnico può avere un impiego anche in più di uno dei centri sportivi. Un centro sportivo può avere uno o più impiegati
impiego	Centro Sportivo	Personale Amministrativo	Quale membro del personale amministrativo ha un impiego in un determinato centro sportivo	Un membro del personale amministrativo può avere un impiego in uno solo dei centri sportivi. Un centro sportivo può avere uno o più impiegati
competenza	Personale Tecnico	Sport	Quale membro del personale tecnico ha competenze in un determinato sport	Un membro del personale tecnico può avere competenze in uno o più sport. Di uno sport possono occuparsi zero o più membri
frequenta	Utente	Centro Sportivi	Quale utente frequenta quale centro sportivo	Un utente può frequentare un solo centro sportivo. Ovviamente un centro sportivo può essere frequentato da più utenti
effettuazione	Utente	Recensione	Quale utente ha effettuato quale recensione	Un utente può effettuare una o più recensioni sul centro sportivo che frequenta
convenzione	Centro Sportivo	Centro Sanitario	Quale dei centri sanitari è convenzionato con uno o più dei centri sportivi e a quando	Un centro sanitario può avere una convenzione con uno o più dei centri sportivi. Un centro sportivo può essere convenzionato

			risale la convenzione	con uno o più dei centri sanitari
convenzione	Centro Sportivo	Centro Donazione	Quale dei centri donazione è convenzionato con uno o più dei centri sportivi e a quando risale la convenzione	Un centro donazione può avere una convenzione con uno o più dei centri sportivi. Un centro sportivo può essere convenzionato con uno o più dei centri donazione
effettuazione	Iscritto	Donazione	Quale iscritto effettua una o più donazioni	Zero o più iscritti possono effettuare una o più donazioni
registro	Donazioni	Centro Donazione	Quale donazione è stata effettuata quando e in quale centro donazioni	In un centro donazioni si possono effettuare zero o più donazioni
effettuazione	Iscritto	Visita	Quale iscritto ha effettuato quando e quale visita	Zero o più iscritti possono effettuare una o più visite
impiego	Centro Sanitario	Specialista	Quale specialista lavora in quale centro sanitario	Uno specialista può lavorare in un solo centro sanitario. In un centro sanitario possono lavorare uno o più specialisti
visitante	Visita	Specialista	Quale specialista ha effettuato quale visita	Una visita può essere effettuata da un solo specialista
locazione	Visita	Centro Sanitario	Quale visita è stata effettuata in quale centro sanitario	In un centro sanitario si possono effettuare zero o più visite

Modifiche e Normalizzazione

Lo schema ER-logico non ci fornisce un modello tramite il quale passare direttamente all'implementazione. Abbiamo quindi bisogno dello schema ER-fisico che otteniamo attraverso varie modifiche alla versione logica.

Queste modifiche sono volte a migliorare quella che sarà la qualità della base di dati. Alcuni aspetti relativi alla qualità sui quali ci si può soffermare sono l'atomicità degli attributi e la presenza delle ridondanze.

La Prima Forma Normale prevede che i valori associati ad un determinato attributo debbano essere atomici dal punto di vista semantico.

Ad esempio consideriamo l'attributo indirizzo presente nelle entità "Centro Sportivo", "Iscritto", "Centro Sanitario" e "Centro Donazione". Abbiamo che per come sarà definito nel dizionario delle entità (in seguito) un indirizzo comprende via/piazza e numero civico. Secondo la Prima Forma Normale si dovrebbe scomporre questo attributo in due attributi dove il primo rappresenterebbe via/piazza e il secondo il numero civico.

Ma nel database che si vuole creare, scomporre questo attributo non è un'azione particolarmente significativa, poiché non c'è motivo di voler conoscere un numero civico separatamente dalla via/piazza. Inoltre le quattro entità verrebbero appesantite da un attributo in più.

Per quanto riguarda il secondo aspetto abbiamo che la progettazione concettuale è stata fatta in maniera tale da minimizzare la presenza di dati ridondanti.

Ma possiamo eliminare la relazione "locazione" che vede collegarsi le entità "Visita" e "Centro Sanitario". Come spiegato precedentemente questa relazione consiste nel tener presente di quale sia il centro sanitario in cui è stata effettuata una determinata visita. Possiamo però ricavare questa informazione anche attraverso la relazione "visitante" tra le entità "Visita" e "Specialista". A partire da una visita si può dunque risalire allo specialista che l'ha effettuata e con la relazione "impiego" risalire al centro sanitario in cui questo lavora.

Eliminiamo quindi la relazione "locazione".

Ci sono anche gli attributi sesso ed età in "Utente" che in qualche modo rappresentano una ridondanza. Questi due dati possono essere infatti derivati dall'entità "Iscritto" tramite la relazione "corrisponde". In "Iscritto" ci sono infatti gli attributi "dataNascita" e "sesso". Tuttavia a fini del progetto si preferisce mantenere queste informazioni anche in "Utenti".

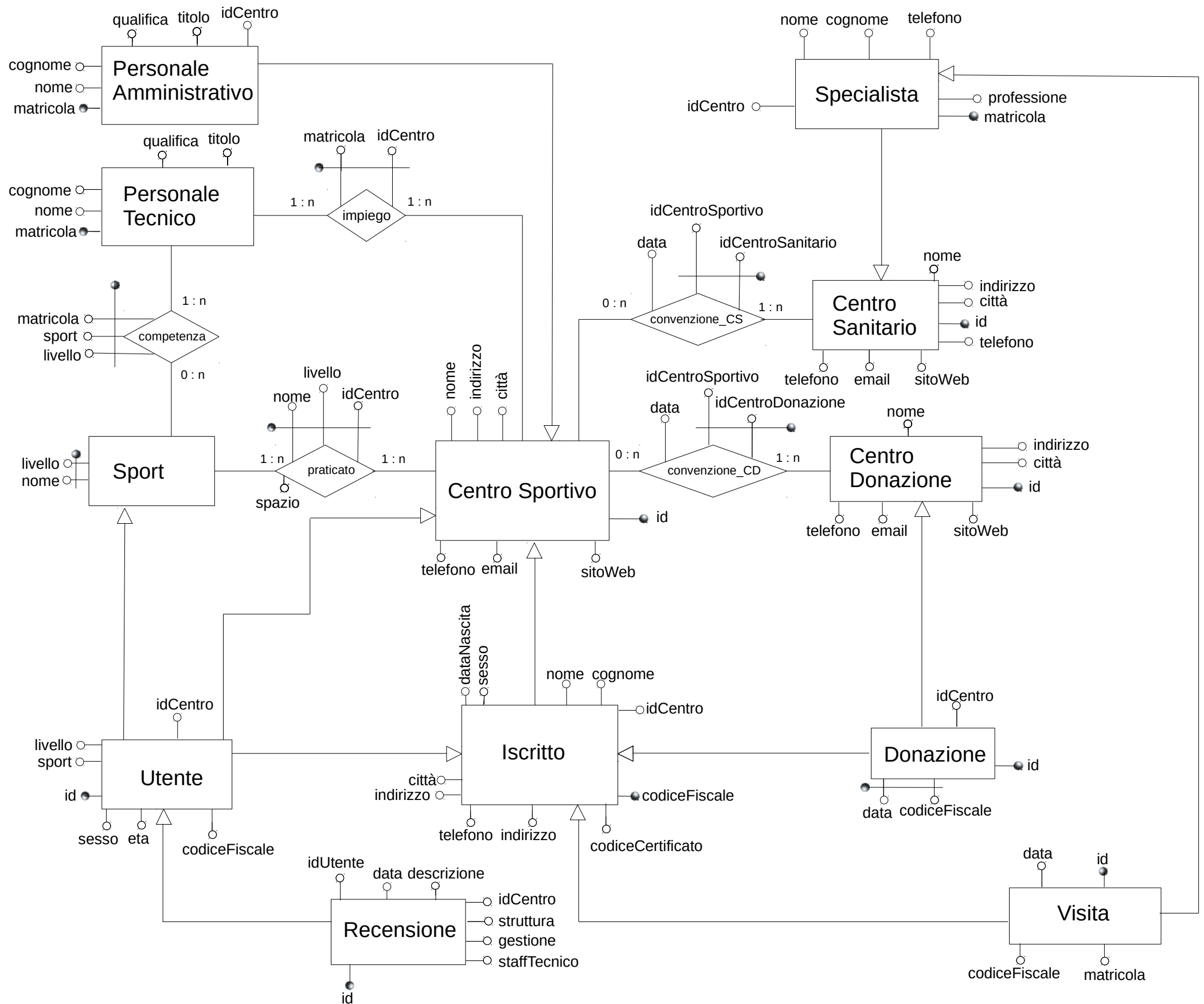
Accade analogamente per "idCentro" in "Recensione". Tramite la relazione "effettuazione" si può risalire al centro sportivo frequentato dall'utente e su cui è stata fatta la recensione. Seppure ciò rappresenti in qualche modo una ridondanza si preferisce avere nell'entità "Recensione" l'attributo "idCentro". Qualora si volesse studiare i record della tabella "Recensione" potrebbe risultare scomodo dover ogni volta andare a ricavare il centro sportivo tramite l'utente.

Per passare allo schema ER-fisico si trasformano le relazioni con cardinalità di tipo 0/1:1 e 0/1:n. Verranno quindi raffigurate con

una freccia rivolta nel verso in cui la cardinalità è 0/1:n. L'entità che invece ha nella relazione cardinalità 0/1:1 ha come chiave esterna la chiave primaria dell'altra entità. Tutte le altre relazioni avranno come chiave primaria una superchiave data dall'unione delle chiavi primarie delle entità che collegano.

Le due relazioni omonime "convenzione" cambiano nome in "convenzione_CS" tra "Centro Sportivo" e "Centro Sanitario" e "convenzione_CD" tra "Centro Sportivo" e "Centro Donazione".

Schema ER-fisico



Entità

Centro Sportivo

Entità che contiene le informazioni generali dei centri sportivi che aderiscono al progetto.

attributo	tipo di dato	descrizione
id(PRIMARY KEY)	INT	Numero che identifica univocamente un centro sportivo
nome	VARCHAR	Il nome del centro sportivo
indirizzo	VARCHAR	Piazza/via e numero civico in cui si trova il centro sportivo
citta	VARCHAR	Città in cui si trova il centro sportivo
telefono	VARCHAR	Numero di telefono del Centro Sportivo
email	VARCHAR	Indirizzo email del centro sportivo
sitoWeb	VARCHAR	Indirizzo del sito Web del centro sportivo

*in fase di implementazione si deciderà la dimensione dei varchar

Iscritto

Entità che descrive l'insieme degli iscritti di ogni centro sportivo presente nell'entità Centro Sportivo.

attributo	tipo di dato	descrizione
codiceFiscale(PRIMARY KEY)	VARCHAR	Codice Fiscale dell'iscritto.Lo identifica univocamente
nome	VARCHAR	Nome proprio dell'iscritto
cognome	VARCHAR	Cognome dell'iscritto
sex	ENUM('maschile','femminile')	Sex dell'iscritto
dataNascita	DATE	Data di nascita dell'iscritto
indirizzo	VARCHAR	Piazza/Via e numero civico in cui risiede l'iscritto
citta	VARCHAR	Città in cui risiede l'iscritto
telefono	VARCHAR	Numero di telefono fisso dell'iscritto
cellulare	VARCHAR	Numero di cellulare dell'iscritto
email	VARCHAR	Indirizzo email

		dell'iscritto
codiceCertificato	VARCHAR	Codice che consente di risalire al certificato medico dell'iscritto(spiegato meglio precedentemente)
idCentro(FOREIGN KEY)	INT	Identificativo del centro sportivo dell'iscritto.Si ricorda che si può frequentare più centri sportivi ma essere iscritti solo ad uno

*idCentroSportivo è una chiave esterna che fa riferimento alla chiave primaria di "Centro Sportivo".

*Le Enum che sono un tipo di dato che rappresenta un insieme di valori fissato sono utilizzate per determinati attributi come sesso,titolo di studio,qualifica ...Questo insieme può essere successivamente ampliato

Sport

Entità che elenca l'insieme degli sport e il livello con cui essi sono praticati nei centri sportivi che aderiscono al progetto.

attributo	tipo di dato	descrizione
nome(PRIMARY KEY)	VARCHAR	Nome dello sport
livello(PRIMARY KEY)	ENUM('professionistico', 'dilettantistico')	Livello con cui lo sport è praticato

*la coppia di attributi "nome" e "livello" forma una chiave primaria.Quindi un'attività praticata nel centro sportivo è identificata univocamente con il nome dello sport e il livello con cui esso è praticato.Quindi potremmo dire che questa entità considera due attività che consistono nello stesso sport,ma praticato in due livelli diversi(quindi agonistico e dilettantistico) come due sport differenti.Il nome dell'entità è dal punto di vista semantico leggermente improprio.

Utente

Entità che contiene tutti gli utenti che praticano un determinato sport ad un determinato livello in un determinato centro.Si ricorda che un iscritto può essere rappresentato da più utenti se questi praticano uno sport in centri diversi.

L'utente rappresenta l'iscritto come praticante(dilettante o agonista) di uno sport in un centro sportivo.

attributo	tipo di dato	descrizione
id(PRIMARY KEY)	BIGINT	Numero che identifica univocamente l'utente
codiceFiscale(FOREIGN KEY)	VARCHAR	Codice Fiscale dell'utente che rappresenta l'iscritto
idCentro(FOREIGN KEY)	INT	Centro sportivo in cui l'utente pratica lo sport
sport(FOREIGN KEY)	VARCHAR	Nome dello sport praticato dall'utente
livello(FOREIGN KEY)	SET('professionistico',	Livello con cui l'utente

	'dilettantistico')	pratica lo sport(dilettantistico, agonistico...)
sexso	ENUM('maschile','femminile')	Sexso dell'utente
eta	TINYINT	Età dell'utente

Personale Tecnico

Entità che rappresenta l'insieme dei membri del personale tecnico dei vari centri sportivi. Un membro del personale tecnico può lavorare in più di uno dei centri sportivi e può insegnare più di uno sport a più di un livello.

attributo	tipo di dato	descrizione
matricola(PRIMARY KEY)	BIGINT	Numero che identifica univocamente un membro del personale tecnico
nome	VARCHAR	Nome del membro del personale tecnico
cognome	VARCHAR	Cognome del membro del personale tecnico
qualifica	ENUM('istruttore', 'preparatore')	Ruolo che ricopre il membro del personale tecnico
titolo	ENUM('laurea', 'diploma')	Titolo di studio del membro del personale tecnico.

Personale Amministrativo

Entità che rappresenta l'insieme dei membri del personale amministrativo dei centri sportivi. Un membro del personale amministrativo che può essere un segretario, un dirigente... può lavorare solo in uno dei centri sportivi (accade diversamente per il personale amministrativo).

attributo	tipo di dato	descrizione
matricola(PRIMARY KEY)	BIGINT	Numero che identifica univocamente un membro del personale amministrativo
idCentro(FOREIGN KEY)	INT	Centro sportivo in cui lavora il membro del personale amministrativo
nome	VARCHAR	Nome del membro del personale amministrativo
cognome	VARCHAR	Cognome del membro del personale amministrativo
qualifica	ENUM('segretario', 'dirigente')	Ruolo che ricopre il membro del personale amministrativo
titolo	ENUM('laurea', 'diploma')	Titolo di studio del membro del personale amministrativo.

Recensioni

Entità che contiene le recensioni fatte dall'utente sul centro sportivo che frequentano. Le recensioni sono caratterizzate da un insieme di voti attribuiti a determinati aspetti di ogni centro sportivo. I voti sono interi compresi tra 1 e 5.

attributo	tipo di dato	descrizione
id(PRIMARY KEY)	BIGINT	Numero che identifica univocamente la recensione
idUtente(FOREIGN KEY)	BIGINT	Id dell'utente che ha effettuato la recensione
idCentro	INT	Id del centro sportivo di cui è effettuata la recensione
data	DATE	Data di effettuazione della recensione
struttura	ENUM(1,2,3,4,5)	Voto attribuito alla struttura
gestione	ENUM(1,2,3,4,5)	Voto attribuito alla gestione
staffTecnico	ENUM(1,2,3,4,5)	Voto attribuito allo staffTecnico
descrizione	TINYTEXT	Testo aggiuntivo relativo alla recensione. Non è obbligatorio

Centro Sanitario

Entità che elenca i centri sanitari convenzionati con uno o più dei centri sportivi. Sarà la relazione convenzione_CS a specificare quali centri sportivi sono convenzionati con quali centri sanitari.

attributo	Tipo di dato	descrizione
id	BIGINT(PRIMARY KEY)	Numero che identifica univocamente il centro sanitario
nome	VARCHAR	Nome del centro sanitario
indirizzo	VARCHAR	Piazza/via e numero civico in cui si trova il centro sanitario
città	VARCHAR	Città in cui si trova il centro sanitario
telefono	VARCHAR	Numero di telefono del Centro sanitario
email	VARCHAR	Indirizzo email del centro sanitario
sitoWeb	VARCHAR	Indirizzo del sito Web del centro sanitario

Specialista

Entità che contiene gli specialisti che lavorano nei vari centri

sanitari. Uno specialista può lavorare in un solo centro sanitario.

attributo	tipo di dato	attributo
matricola(PRIMARY KEY)	VARCHAR	Numero che identifica univocamente lo specialista
professione	VARCHAR	Professione dello specialista. Può essere un medico di base, un ortopedico, un fisioterapista...
nome	VARCHAR	Nome dello specialista
cognome	VARCHAR	Cognome dello specialista
idCentro(FOREIGN KEY)	INT	Id del centro sanitario in cui lavora lo specialista
telefono	VARCHAR	Numero di telefono dello specialista
email	VARCHAR	Indirizzo email dello specialista

Visita

Entità che registra le visite effettuate dagli iscritti presso uno degli specialisti che lavora in un centro sanitario convenzionato.

attributo	Tipo di dato	descrizione
id(PRIMARY KEY)	BIGINT	Numero che identifica univocamente la visita
codiceFiscale(FOREIGN KEY)	VARCHAR	Codice Fiscale dell'iscritto che effettua la visita
matricolaSpecialista(FOREIGN KEY)	BIGINT	Matricola dello specialista che esegue la visita
data	DATE	Data in cui è effettuata la visita

Centro Donazione

Entità che elenca i centri in cui si può donare il sangue convenzionati con uno o più dei centri sportivi. Sarà la relazione convenzione_CD a specificare quali centri sportivi sono convenzionati con quali centri donazione.

attributo	Tipo di dato	descrizione
id(PRIMARY KEY)	BIGINT	Numero che identifica univocamente il centro donazione
nome	VARCHAR	Nome del centro sanitario
indirizzo	VARCHAR	Piazza/via e numero civico in cui si trova il centro donazione
citta	VARCHAR	Città in cui si trova il

		centro donazione
telefono	VARCHAR	Numero di telefono del centro donazione
email	VARCHAR	Indirizzo email del centro donazione
sitoWeb	VARCHAR	Indirizzo del sito Web del centro donazione

Donazione

Entità che registra le donazioni effettuate dagli iscritti presso uno dei centri donazione convenzionato con uno o più dei centri sportivi.

attributo	Tipo di dato	descrizione
id(PRIMARY KEY)	BIGINT	Numero che identifica univocamente la donazione
codiceFiscale(FOREIGN KEY)	VARCHAR	Codice Fiscale dell'iscritto che effettua la donazione
data	DATE	Data in cui è effettuata la donazione
idCentro(FOREIGN KEY)	INT	Id del centro(donazione) in cui è stata effettuata la donazione

Relazioni

praticato

Relazione che tiene traccia di quale sport a quale livello è praticato in quali dei centri sportivi e lo spazio in cui questo è praticato(indoor o outdoor).

attributo	Tipo di dato	descrizione
sport(PRIMARY KEY)	VARCHAR	Nome dello sport praticato
livello(PRIMARY KEY)	ENUM('professionistico', 'dilettantistico')	Livello con cui lo sport è praticato
IdCentro(PRIMARY KEY)	INT	Id del centro(donazione) in cui è praticato lo sport a un determinato

		livello
spazio(PRIMARY KEY)	ENUM('indoor', 'outdoor', 'entrambi')	Tipologia di spazio in cui lo sport è praticato. Indoor,Outdoor o entrambi

competenza

Relazione che tiene traccia di quale membro del personale tecnico si occupa di un determinato sport a un determinato livello.

attributo	Tipo di dato	descrizione
matricola(PRIMARY KEY)	BIGINT	Matricola del membro del personale tecnico
sport(PRIMARY KEY)	VARCHAR	Nome dello sport praticato
livello(PRIMARY KEY)	ENUM('professionistico', 'dilettantistico')	Livello con cui lo sport è praticato

impiego

Relazione che tiene traccia di quale membro del personale tecnico lavora in quali dei centri sportivi.

attributo	Tipo di dato	descrizione
matricola(PRIMARY KEY)	BIGINT	Matricola del membro del personale tecnico
idCentro(PRIMARY KEY)	INT	Id del centro sportivo in cui lavora il membro del personale tecnico

convenzione_CS

Relazione che tiene traccia di quali centri sanitari sono convenzionati con quali dei centri sportivi.

attributo	Tipo di dato	descrizione
idCentroSportivo(PRIMARY KEY)	INT	Numero che identifica univocamente il centro sportivo
idCentroSanitario(PRIMARY KEY)	BIGINT	Numero che identifica univocamente il centro sanitario
data	DATE	Data in cui è stata stipulata la convenzione

convenzione_CD

Relazione che tiene traccia di quali centri donazione sono convenzionati con quali dei centri sportivi.

attributo	Tipo di dato	descrizione
idCentroSportivo(PRIMARY KEY)	INT	Numero che identifica univocamente il centro sportivo
idCentroDonazione(PRIMARY KEY)	BIGINT	Numero che identifica univocamente il centro donazione
data	DATE	Data in cui è stata effettuata la donazione

CREATE ED INSERT

CentroSportivo

```
create table CentroSportivo(
    id INT AUTO_INCREMENT,
    nome VARCHAR(30) NULL,
    indirizzo VARCHAR(100) NOT NULL,
    citta VARCHAR(20) NOT NULL,
    telefono VARCHAR(15) NOT NULL UNIQUE,
    email VARCHAR(50) NOT NULL,
    sitoWeb VARCHAR(40) NULL,
    PRIMARY KEY(id)
) ENGINE=InnoDB;

insert into CentroSportivo values
(null,"Sport Center","via Italo
869","Roma","068609452","SportCenter@hotmail.com","www.SportCenter.it"),
(null,"Sportessere","via Carducci
45","Frosinone","068609487","Sportessere@tin.com","www.Sportessere.it"),
(null,"Svago & Forma","piazza Repubblica
78","Velletri","0894609472","Svago.Forma@gmail.com","SvagoForma.it"),
(null,"Atletigame","via dei Draghi
345","Roma","088609463","Atletigame@libero.it","Atletigame.com"), (null,"Country
Club","via Pozzi
34","Napoli","089306789","countryclub@wind.com","www.CountryClubCentro.com"),
(null,"Salerno Sport","via Vietri
56","Salerno","089407123","centrosportivo.salerno56@outlook.com",null),
(null,"Lucen Sport","via Conte Max
123","Roma","088678574","lucensport@hotmail.com","www.lucensport.it"),
(null,"Appia Sport Village","via Appia Moderna
777","Roma","088645328","appiasport.village@gmail.it","www.appiasport.com");
```

Iscritto

```
create table Iscritto(
    codiceFiscale VARCHAR(20),
    idCentro INT NOT NULL,
    nome VARCHAR(20) NOT NULL,
    cognome VARCHAR(30) NOT NULL,
    sesso ENUM("maschile","femminile") NOT NULL,
    dataNascita DATE NOT NULL,
    indirizzo VARCHAR(100) NOT NULL,
    citta VARCHAR(20) NOT NULL,
    telefono VARCHAR(15) NOT NULL UNIQUE,
    email VARCHAR(50) NULL,
    codiceCertificato INT UNIQUE NULL,
    PRIMARY KEY(codiceFiscale),
    FOREIGN KEY (idCentro) references CentroSportivo(id)
        ON UPDATE CASCADE
        ON DELETE CASCADE
) ENGINE=InnoDB;

insert into Iscritto values
("RSSRRT56C61H501C",1,"Roberta","Rossi","femminile","1956-03-21","via Casali
12","Roma","06869589","robertarossi@gmail.it",1523),
("FRRMRC99D03D810Y",2,"Marco","Ferri","maschile","1999-04-03","via Veneti
121","Frosinone","06859581","marcoferri@gmail.it",18294),
("DSTLCU02E49L719G",3,"Lucia","Destro","femminile","2002-05-09","via Argentina
126","Velletri","06849723","luciadestro@libero.it",89982),
("DSNPLA75R01H501K",8,"Paolo","De Santis","maschile","1975-10-01","via Tuscolo
99","Roma","06869526","paolodesantis@wind.it",17784),
```



```
( "STNNDR92D58H703R",6,"Andrea","Stankovic","femminile","1992-04-18","via Reni
15","Amalfi","06459832","andreastankovic@wind.it",2524),
( "VLTGNN93D23A509S",5,"Giovanni","Veltri","maschile","1993-04-23","piazza Caruso
4","Napoli","06969421","giovanniveltri@tin.com",68524),
( "BRRRCC70C50G482B",7,"Rebecca","Barry","femminile","1970-03-10","via Regina
Elisabetta 12","Roma","06867478","rebeccabarry@gmail.com",134524),
( "CHRSF005S52H501J",1,"Sofia","Chiari","femminile","2005-11-12","via Italo
560","Roma","06863521","sofiachiari@hotmail.it",98524),
( "RSSJPL67A30F205H",1,"Jean Paul","Rossi","maschile","1967-01-30","via Armenia
120","Roma","06869521","jeanpaulrossi@gmail.com",44524),
( "VRDMSM79L05F839F",8,"Massimo","Verdini","maschile","1979-07-05","via Appia
Vecchia 345","Roma","06869651","massimoverdini@outlook.com",65524),
( "PSCDGI83P09H703Z",5,"Diego","Pascoli","maschile","1983-09-09","via dei
Cardinali 126","Caserta","06959176","diegopascoli0909@gmail.com",1536),
( "RMOFNC65M12F839U",5,"Francesco","Roma","maschile","1965-08-12","via De Curtis
222","Napoli","069667213",null,15654),
( "TRPDRA49B19G273G",4,"Dario","Trapani","maschile","1949-02-19","piazza
Risorgimento 126","Roma","06865423","dariotrapani@gmail.com",8924),
( "CRTLSN73C41A345V",7,"Alessandra","Cortesi","femminile","1973-03-01","via
dell'Angelo 45","Roma","06860023","alessandracortesi@libero.it",54324),
( "RMPTZN94H53D969B",1,"Tiziana","Rampini","maschile","1979-07-05","via del
Gladiatore 540","Roma","06869001","tizianarampini@hotmail.it",78124),
( "FNTMRC03T24H501H",4,"Marco","Fanti","maschile","2003-12-24","via Lucio Fontana
34","Roma","06869505","marcofanti03@gmail.com",9866),
( "PRNGCM00D24D612C",6,"Giacomo","Parini","maschile","2000-04-24","via Brega
93","Roma","06869771",null,1524);
```

Sport

```
create table Sport(
    nome VARCHAR(30) NOT NULL,
    livello ENUM("professionistico","dilettantistico") NOT NULL,
    PRIMARY KEY(nome,livello)
) ENGINE=InnoDB;
```

```
insert into Sport values
("calcio","dilettantistico"),
("calcio","professionistico"),
("tennis","dilettantistico"),
("tennis","professionistico"),
("pallavolo","dilettantistico"),
("pallavolo","professionistico"),
("pallacanestro","dilettantistico"),
("pallacanestro","professionistico"),
("rugby","dilettantistico"),
("rugby","professionistico"),
("nuoto","dilettantistico"),
("nuoto","professionistico"),
("pallanuoto","dilettantistico"),
("pallanuoto","professionistico"),
("golf","dilettantistico"),
("golf","professionistico"),
("pugilato","dilettantistico"),
("pugilato","professionistico");
```

Utente

```
create table Utente(  
    id BIGINT AUTO_INCREMENT,  
    codiceFiscale VARCHAR(20) NOT NULL,  
    idCentro INT NOT NULL,  
    sport VARCHAR(30) NOT NULL,  
    livello ENUM("professionistico","dilettantistico") NOT NULL,  
    sesso ENUM("maschile","femminile") NOT NULL,  
    eta INT UNSIGNED NOT NULL,  
    PRIMARY KEY(id),  
    FOREIGN KEY (codiceFiscale) references Iscritto(codiceFiscale)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    FOREIGN KEY (idCentro) references CentroSportivo(id)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    FOREIGN KEY (sport,livello) references Sport(nome,livello)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
) ENGINE=InnoDB;
```

/*Gli inserimenti in Utente verranno effettuati in un secondo momento.
Si osserva che in Utente sono presenti i campi "sesso" ed "eta" che essendo derivabili da Iscritto rappresentano in qualche modo una ridondanza.
Ma per motivi di compattezza e per gli scopi dell'applicazione si vogliono avere le informazioni replicate in Utente. Se ad esempio si volessero conoscere tutti gli utenti di una certa età bisognerebbe fare una query più semplice e lineare piuttosto di una fatta eliminando le ridondanze.
Ciò ha un prezzo e cioè che bisogna controllare che sesso ed età in Utente coincidano con le informazioni in Iscritto.
Creiamo dunque due trigger che effettuano i controlli prima dell'inserimento*/
IN SEGUITO VERRANNO MOSTRATI I DUE TRIGGER E LE INSERTS

PersonaleTecnico

```
create table PersonaleTecnico(  
    matricola BIGINT AUTO_INCREMENT,  
    nome VARCHAR(20) NOT NULL,  
    cognome VARCHAR(30) NOT NULL,  
    qualifica ENUM("istruttore","preparatore") NOT NULL,  
    titolo ENUM("diploma","laurea"),  
    PRIMARY KEY(matricola)  
) ENGINE=InnoDB;
```

```
insert into PersonaleTecnico values  
(null,"Gabriele","Zito","istruttore","diploma"),  
(null,"Biagio","Napolitani","istruttore","laurea"),  
(null,"Giuditta","Fallaci","preparatore","diploma"),  
(null,"Loredana","Conti","istruttore","diploma"),  
(null,"Francesca","Cattaneo","istruttore",null),  
(null,"Alessandra","Pinto","preparatore","laurea"),  
(null,"Fernando","Pugliesi","istruttore","diploma"),  
(null,"Raffaele","Costa","istruttore","laurea"),  
(null,"Silvia","Trevisano","preparatore","diploma"),  
(null,"Augusto","Fiorentini","preparatore","diploma"),  
(null,"Emanuele","Ferrara","istruttore",null),  
(null,"Fabrizio","Milani","istruttore","diploma");
```

PersonaleAmministrativo

```
create table PersonaleAmministrativo(  
    matricola BIGINT AUTO_INCREMENT,  
    idCentro INT NOT NULL,  
    nome VARCHAR(20) NOT NULL,  
    cognome VARCHAR(30) NOT NULL,  
    qualifica ENUM("segretario","dirigente") NOT NULL,  
    titolo ENUM("diploma","laurea"),  
    PRIMARY KEY(matricola),  
    FOREIGN KEY (idCentro) references CentroSportivo(id)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
) ENGINE=InnoDB;
```

```
insert into PersonaleAmministrativo values  
(null,6,"Elisabetta","Manna","dirigente","laurea"),  
(null,8,"Flaviana","Toscani","segretario","diploma"),  
(null,1,"Giacomo","Milani","dirigente","diploma"),  
(null,2,"Amerigo","Giordano","dirigente","laurea"),  
(null,4,"Carolina","Gallo","segretario",null),  
(null,7,"Aldo","Trevisan","segretario","laurea"),  
(null,3,"Gianfranco","Bicchi","dirigente","laurea"),  
(null,5,"Stefania","Ruzic","segretario",null),  
(null,8,"Vincenzo","Petrangeli","dirigente","diploma"),  
(null,7,"Christian","Carli","segretario","diploma"),  
(null,4,"Ivan","Borletti","dirigente",null),  
(null,3,"Pierluigi","Michelin","segretario","laurea");
```

Recensione

```
create table Recensione(  
    id BIGINT AUTO_INCREMENT,  
    idUtente BIGINT NOT NULL,  
    idCentro INT,  
    data DATE NOT NULL,  
    struttura ENUM("1","2","3","4","5") NOT NULL,  
    gestione ENUM("1","2","3","4","5") NOT NULL,  
    staffTecnico ENUM("1","2","3","4","5") NOT NULL,  
    PRIMARY KEY(id),  
    FOREIGN KEY (idUtente) references Utente(id)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
) ENGINE=InnoDB;
```

Gli inserimenti verranno mostrati in seguito.

Prima di effettuare gli inserimenti si vuole creare un trigger che faccia sì che si inserisca in automatico l'id del centro sportivo dell'utente che effettua la recensione.

ricorda trigger per inserire id del centro sportivo dell'utente

****INSERT SULLA TABELLA RECENSIONE MOSTRATE IN SEGUITO****

CentroSanitario

```
create table CentroSanitario(  
    id BIGINT AUTO_INCREMENT,  
    nome VARCHAR(30) NULL,  
    indirizzo VARCHAR(100) NOT NULL,  
    citta VARCHAR(20) NOT NULL,
```

```

    telefono VARCHAR(15) NOT NULL UNIQUE,
    email VARCHAR(50) NOT NULL UNIQUE,
    sitoWeb VARCHAR(40) NULL UNIQUE,
    PRIMARY KEY(id)
) ENGINE=InnoDB;

insert into CentroSanitario values
(null,"C&V","via Davide
865","Roma","06862826","C&V.centroSanitario@hotmail.com","www.C&Vcentro.it"),
(null,"Centro Pugliese","via Palestro
365","Frosinone","068609487","CentroPugliese@tin.com","www.CentroPugliese.it"),
(null,"Fisiosports","piazza San Rocco
90","Velletri","08947424","Fisiosports@gmail.com","Fisiosports .it"),
(null,"Polispecialistica Rossi","via dei Mattarella
644","Roma","08869731","Polispecialistica Rossi@libero.it","Polispecialistica
Rossi .com"), (null,"Studio Corpore Sano","via Lancellotti
34","Napoli","08930215","CorporeSano@wind.com","www.CorporeSanoCentro.com"),
(null,"Studio Paideia","via Cardinal Franchi
21","Salerno","08948145","StudioPaideia.salerno56@outlook.com",null),
(null,"Almamed","via Boldi
654","Roma","088678574","AlmaMed@hotmail.com","www.almamed.it"),
(null,"Poliambulatorio Cecconi","via Appia Moderna
157","Roma","08867634","poliCecconi@gmail.it","www.poliCecconi.com"),
(null,null,"via Bastille
282","Roma","08862261","studioBastille@gmail.it","www.studioBastille.com"),
(null,"Progetto Salute srl","via Appia Moderna
4","Roma","08865963","obbiettivoSalute@gmail.it","www.ProgettoSalute.com");

```

Specialista

```

create table Specialista(
    matricola BIGINT AUTO_INCREMENT,
    professione VARCHAR(40) NOT NULL,
    idCentro BIGINT NOT NULL,
    nome VARCHAR(20) NOT NULL,
    cognome VARCHAR(30) NOT NULL,
    telefono VARCHAR(15) NOT NULL UNIQUE,
    email VARCHAR(50) NOT NULL,
    PRIMARY KEY(matricola),
    FOREIGN KEY (idCentro) references CentroSanitario(id)
        ON UPDATE CASCADE
        ON DELETE CASCADE
) ENGINE=InnoDB;

insert into Specialista values
(null,"fisioterapista",3,"Luca","Vialli","3235971357","lucavialli.fisio@hotmail.
com"),
(null,"osteopata",1,"Catia","Paletta","3771796846","osteopatacatiapaletta@gmail.
com"),
(null,"medico di base",
5,"Igor","Cristiani","3730893512","dottorigorcristiani@gmail.com"),
(null,"otorino",7,"Michela","Bordini","3451792246","michelabordini@gmail.com"),
(null,"oculista",1,"Andrea","Talamonti","3281794416","C&V.centroSanitario@hotmai
l.com"),
(null,"ortopedico",8,"Chiara","Bizanti","3339824416","ChiaraBizanti@hotmail.com"
),
(null,"fisiatra",10,"Claudio","Berni","069417416","ClaudioBerniFisiatra@gmail.it
"),
(null,"fisioterapista",9,"Alexander","Friedkin","3461749716","FriedkinAlex.phisi
o@outlook.com"),
(null,"oculista",6,"Luana","Perlati","360176516","PerlatiLuanaOculista@libero.co
m"),
(null,"medico di base",7,

```

```
"Anna Maria", "Santino", "067874416", "SantinoMaria@hotmail.com"),
(null, "fisioterapista", 7, "Evaldo", "Patani", "3669896346", "evaldopatani@fastweb.com"),
(null, "osteopata", 1, "Ivana", "De
Rossi", "068997846", "derossiivania.osteo@gmail.com"),
(null, "ortopedico", 4, "Danila", "Linari", "3451796216", "danilalinari@wind.it"),
(null, "psicologo", 2, "Fausto", "Elvar", "069598346", "elvarfausto.psico@tin.com"),
(null, "neurolologo", 4, "Giovanna", "Paltrinieri", "3772368462", "giovannapaltrinieri@gmail.com"),
(null, "psicologo", 5, "Yuri", "Breve", "3281713845", "psicologoyuribreve@hotmail.it"),
',
(null, "pneumologo", 10, "Cristiana", "Arena", "34512334816", "cristianaarena.pneumologia@outlook.com");
```

Visita

```
create table Visita(
    id BIGINT AUTO_INCREMENT,
    codiceFiscale VARCHAR(20) NOT NULL,
    data DATE NOT NULL,
    matricolaSpecialista BIGINT NOT NULL,
    PRIMARY KEY(id),
    FOREIGN KEY (codiceFiscale) references Iscritto(codiceFiscale)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (matricolaSpecialista) references Specialista(matricola)
        ON UPDATE CASCADE
        ON DELETE CASCADE
) ENGINE=InnoDB;
```

```
insert into Visita values
(null, "RMOFNC65M12F839U", "2020-11-19", 19),
(null, "PRNGCM00D24D612C", "2021-05-16", 32),
(null, "RSSJPL67A30F205H", "2019-04-04", 30),
(null, "PSCDGI83P09H703Z", "2021-10-09", 18),
(null, "CRTLSN73C41A345V", "2020-02-22", 25),
(null, "RSSRRT56C61H501C", "2021-01-31", 22),
(null, "BRRRCC70C50G482B", "2019-06-18", 19),
(null, "FNTMRC03T24H501H", "2020-09-26", 27),
(null, "STNNDR92D58H703R", "2020-12-05", 21),
(null, "STNNDR92D58H703R", "2019-11-21", 22);
```

CentroDonazione

```
create table CentroDonazione(
    id BIGINT AUTO_INCREMENT,
    nome VARCHAR(30) NULL,
    indirizzo VARCHAR(100) NOT NULL,
    citta VARCHAR(20) NOT NULL,
    telefono VARCHAR(15) NOT NULL UNIQUE,
    email VARCHAR(50) NOT NULL UNIQUE,
    sitoWeb VARCHAR(40) NULL UNIQUE,
    PRIMARY KEY(id)
) ENGINE=InnoDB;
```

```
insert into CentroDonazione values
(null, "AVIS", "via Golia
85", "Roma", "06862324", "Avis.viagolia@hotmail.com", "www.avisGolia.it"),
(null, "Centro Braschi", "via Dainelli
135", "Frosinone", "068617586", "CentroBraschi@tin.com", "www.BraschiDonazioni.it"),
(null, "Associazione Donatori", "via Episcopo
```

```

450", "Velletri", "08943583", "AssociazioneDonatori@gmail.com", "ADEpiscopo.com"),
(null, "AVIS", "via dei Angiolieri
260", "Roma", "08854731", "Avis.vianagiolieri@libero.it", "AvisAngiolieri.com"),
(null, "Centro EMA", "via Sisto
55", "Napoli", "08934416", "CentroEma@wind.com", "www.EMAdonazioni.com"),
(null, "AVIS", "via Greco 231", "Salerno", "08942247", "AvisGreco@outlook.com", null),
(null, "Centro Nazionale Donazioni", "via Rita
22", "Roma", "088635571", "CentroNazionaleDonazioni@hotmail.com", "www.CNdonazioni.o
rg"),
(null, "AVIS", "via Appia Moderna
40", "Roma", "08868532", "AppiAvis@gmail.it", "www.appiavis.com"),
(null, null, "via Scarselli
146", "Roma", "08862431", "donazioniviaScarselli@gmail.it", "www.donareFabene.it"),
(null, "Progetto EMAS", "via Calvairate
2", "Roma", "08809963", "progettoEMAS@gmail.it", "www.ProgettoEMAS.com");

```

Donazione

```

create table Donazione(
    id BIGINT AUTO_INCREMENT,
    codiceFiscale VARCHAR(20) NOT NULL,
    data DATE NOT NULL,
    idCentro BIGINT NOT NULL,
    PRIMARY KEY(id),
    FOREIGN KEY (codiceFiscale) references Iscritto(codiceFiscale)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (idCentro) references CentroDonazione(id)
        ON UPDATE CASCADE
        ON DELETE CASCADE
) ENGINE=InnoDB;

```

/*Un Iscritto non può donare più di una volta ogni 90 giorni.
Prima di effettuare gli inserimenti occorre creare un trigger che possa impedire
l'inserimento di un record che rappresenta la donazione di un donatore che ha
donato meno di 90 giorni prima*/

IN SEGUITO VERRANNO MOSTRATI IL TRIGGER E LE INSERTS

Praticato

```

create table Praticato(
    sport VARCHAR(30) NOT NULL,
    livello ENUM("professionistico", "dilettantistico") NOT NULL,
    idCentro INT NOT NULL,
    spazio ENUM("indoor", "outdoor") NOT NULL,
    PRIMARY KEY(sport, livello, idCentro, spazio),
    FOREIGN KEY (sport, livello) references Sport(nome, livello)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (idCentro) references CentroSportivo(id)
        ON UPDATE CASCADE
        ON DELETE CASCADE
) ENGINE=InnoDB;

```

```

insert into Praticato values
("calcio", "dilettantistico", 7, "outdoor"),
("calcio", "professionistico", 7, "outdoor"),
("calcio", "professionistico", 6, "outdoor"),
("tennis", "dilettantistico", 5, "outdoor"),
("tennis", "professionistico", 5, "indoor"),
("tennis", "professionistico", 5, "outdoor"),

```

```
(
    "tennis","professionistico",4,"indoor"),
    ("pallavolo","dilettantistico",1,"indoor"),
    ("pallavolo","dilettantistico",8,"indoor"),
    ("pallavolo","professionistico",4,"indoor"),
    ("pallavolo","professionistico",2,"indoor"),
    ("pallacanestro","dilettantistico",2,"indoor"),
    ("pallacanestro","professionistico",2,"indoor"),
    ("rugby","dilettantistico",6,"outdoor"),
    ("rugby","professionistico",6,"outdoor"),
    ("rugby","professionistico",3,"outdoor"),
    ("nuoto","dilettantistico",2,"indoor"),
    ("nuoto","dilettantistico",1,"indoor"),
    ("nuoto","professionistico",2,"indoor"),
    ("pallanuoto","dilettantistico",2,"indoor"),
    ("pallanuoto","professionistico",1,"indoor"),
    ("pallanuoto","professionistico",2,"indoor"),
    ("golf","dilettantistico",7,"outdoor"),
    ("golf","professionistico",7,"outdoor"),
    ("golf","professionistico",5,"outdoor"),
    ("pugilato","dilettantistico",2,"indoor"),
    ("pugilato","dilettantistico",1,"indoor"),
    ("pugilato","professionistico",2,"indoor");
```

Competenza

```
create table Competenza(
    matricola BIGINT NOT NULL,
    sport VARCHAR(30) NOT NULL,
    livello ENUM("professionistico","dilettantistico") NOT NULL,
    PRIMARY KEY(matricola,sport,livello),
    FOREIGN KEY (sport,livello) references Sport(nome,livello)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (matricola) references PersonaleTecnico(matricola)
        ON UPDATE CASCADE
        ON DELETE CASCADE
) ENGINE=InnoDB;
```

```
insert into Competenza values
(3,"calcio","dilettantistico"),
(8,"calcio","dilettantistico"),
(4,"calcio","professionistico"),
(10,"calcio","professionistico"),
(5,"calcio","professionistico"),
(7,"tennis","dilettantistico"),
(7,"tennis","professionistico"),
(6,"tennis","professionistico"),
(11,"tennis","professionistico"),
(9,"pallavolo","dilettantistico"),
(12,"pallacanestro","dilettantistico"),
(12,"pallacanestro","professionistico"),
(10,"rugby","dilettantistico"),
(10,"rugby","professionistico"),
(1,"rugby","professionistico"),
(9,"nuoto","dilettantistico"),
(2,"nuoto","dilettantistico"),
(2,"nuoto","professionistico"),
(2,"pallanuoto","dilettantistico"),
(9,"pallanuoto","professionistico"),
(2,"pallanuoto","professionistico"),
(3,"golf","dilettantistico"),
(3,"golf","professionistico"),
(1,"pugilato","dilettantistico"),
```

```
(5,"rugby","dilettantistico");
```

Impiego

```
create table Impiego(  
    matricola BIGINT NOT NULL,  
    idCentro INT NOT NULL,  
    PRIMARY KEY(matricola,idCentro),  
    FOREIGN KEY (idCentro) references CentroSportivo(id)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    FOREIGN KEY (matricola) references PersonaleTecnico(matricola)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
) ENGINE=InnoDB;
```

```
insert into Impiego values
```

```
(9,1),  
(9,2),  
(4,7),  
(7,5),  
(1,3),  
(2,2),  
(10,6),  
(6,5),  
(3,7),  
(10,3),  
(5,6),  
(1,6),  
(8,7),  
(9,8);
```

Convenzione_CS

```
create table Convenzione_CS(  
    idCentroSanitario BIGINT NOT NULL,  
    idCentroSportivo INT NOT NULL,  
    data DATE,  
    PRIMARY KEY(idCentroSanitario,idCentroSportivo),  
    FOREIGN KEY (idCentroSportivo) references CentroSportivo(id)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    FOREIGN KEY (idCentroSanitario) references CentroSanitario(id)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
) ENGINE=InnoDB;
```

```
insert into Convenzione_CS values
```

```
(4,1,"2020-04-23"),  
(4,8,"2019-11-03"),  
(8,8,"2021-01-10"),  
(8,1,"2020-08-24"),  
(2,2,"2021-05-07"),  
(3,3,"2020-12-01"),  
(7,3,"2019-07-29"),  
(1,4,"2021-03-18"),  
(1,8,"2020-10-30"),  
(5,5,"2021-02-05"),  
(5,6,"2020-09-21"),  
(6,6,"2020-04-22"),  
(10,1,"2020-08-16"),
```



```
(9,4,"2020-05-08"),  
(7,1,"2019-06-03");
```

Convenzione_CD

```
create table Convenzione_CD(  
    idCentroDonazione BIGINT NOT NULL,  
    idCentroSportivo INT NOT NULL,  
    data DATE,  
    PRIMARY KEY(idCentroDonazione,idCentroSportivo),  
    FOREIGN KEY (idCentroSportivo) references CentroSportivo(id)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    FOREIGN KEY (idCentroDonazione) references CentroDonazione(id)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
) ENGINE=InnoDB;
```

```
insert into Convenzione_CD values  
(1,8,"2021-05-12"),  
(2,2,"2020-05-27"),  
(2,3,"2020-11-03"),  
(3,3,"2019-08-05"),  
(4,1,"2020-08-09"),  
(5,6,"2020-10-28"),  
(5,5,"2021-03-20"),  
(6,6,"2021-01-15"),  
(7,1,"2019-12-01"),  
(7,2,"2019-04-30"),  
(10,1,"2020-11-26"),  
(8,4,"2019-10-18"),  
(9,8,"2019-11-16");
```

TRIGGERS

Seppure rappresenti una ridondanza, nella tabella Utente è presente il campo "eta". L'età è derivabile dalla data di nascita presente nella tabella Iscritto.

Per avere le informazioni utili relative all'utente compattate in una sola tabella, vi sono inseriti campi il cui valore è ricavabile da Iscritto.

Stessa cosa infatti accade per sesso=("maschile","femminile")

Il seguente trigger verifica che l'età inserita nell'utente sia corretta rispetto alla differenza tra la sua data di nascita e la data corrente prima dell'inserimento di un record.

```
delimiter !!
DROP TRIGGER IF EXISTS checkAge;
CREATE TRIGGER checkAge
BEFORE INSERT ON Utente
FOR EACH ROW
BEGIN
DECLARE birthDay DATE;
DECLARE yearsOld INT UNSIGNED;
SET birthDay = (SELECT dataNascita FROM Iscritto WHERE Iscritto.codiceFiscale =
NEW.codiceFiscale);
SET yearsOld = (select
CAST(DATE_FORMAT(FROM_DAYS(DATEDIFF(now(),birthDay)),'%Y') AS UNSIGNED));
IF NEW.eta != yearsOld THEN
SET NEW.eta = yearsOld;
END IF;
END !!
delimiter ;
```

Il seguente trigger verifica che il sesso inserito nell'utente sia corretto rispetto a quanto scritto nel record corrisponde nella tabella Iscritto prima dell'inserimento.

```
delimiter !!
DROP TRIGGER IF EXISTS checkSesso;
CREATE TRIGGER checkSesso
BEFORE INSERT ON Utente
FOR EACH ROW
BEGIN
DECLARE sex ENUM("maschile","femminile");
SET sex = (SELECT Iscritto.sesso FROM Iscritto WHERE Iscritto.codiceFiscale =
NEW.codiceFiscale);
IF NEW.sesso != sex THEN
SET NEW.sesso = sex;
END IF;
END !!
delimiter ;
```

INSERTS SULLA TABELLA UTENTE

```
insert into Utente values
(null,"RSSRRT56C61H501C",1,"tennis","professionistico","femminile",66),
(null,"RSSRRT56C61H501C",3,"nuoto","dilettantistico","femminile",66),
(null,"FRMRRC99D03D810Y",2,"calcio","professionistico","maschile",23),
(null,"FRMRRC99D03D810Y",2,"pallacanestro","dilettantistico","maschile",23),
(null,"FRMRRC99D03D810Y",4,"pallanuoto","professionistico","maschile",23),
(null,"DSTLCU02E49L719G",3,"calcio","professionistico","femminile",20),
(null,"DSNPLA75R01H501K",8,"pallavolo","professionistico","maschile",47),
(null,"DSNPLA75R01H501K",7,"pallanuoto","dilettantistico","maschile",47),
```

```
(null,"STNNDR92D58H703R",6,"tennis","professionistico","femminile",30),
(null,"STNNDR92D58H703R",4,"nuoto","professionistico","femminile",66),
(null,"STNNDR92D58H703R",2,"tennis","professionistico","femminile",66),
(null,"VLTGNN93D23A509S",5,"tennis","professionistico","maschile",29),
(null,"VLTGNN93D23A509S",6,"pugilato","dilettantistico","maschile",29),
(null,"BRRRCC70C50G482B",7,"pallavolo","dilettantistico","femminile",52),
(null,"CHRSF005S52H501J",1,"pallavolo","professionistico","femminile",17),
(null,"CHRSF005S52H501J",7,"pallavolo","professionistico","femminile",17),
(null,"CHRSF005S52H501J",7,"pallanuoto","professionistico","femminile",17),
(null,"RSSJPL67A30F205H",1,"nuoto","dilettantistico","maschile",55),
(null,"RSSJPL67A30F205H",5,"nuoto","dilettantistico","maschile",55),
(null,"VRDMSM79L05F839F",8,"pugilato","dilettantistico","maschile",43),
(null,"VRDMSM79L05F839F",6,"pugilato","dilettantistico","maschile",43),
(null,"PSCDGI83P09H703Z",5,"pallacanestro","dilettantistico","maschile",39),
(null,"RMOFNC65M12F839U",5,"pallacanestro","dilettantistico","maschile",57),
(null,"RMOFNC65M12F839U",5,"pallavolo","dilettantistico","maschile",57),
(null,"TRPDRA49B19G273G",4,"pugilato","dilettantistico","maschile",73),
(null,"CRTLSN73C41A345V",7,"pugilato","dilettantistico","femminile",49),
(null,"RMPTZN94H53D969B",1,"pugilato","professionistico","femminile",43),
(null,"FNTMRC03T24H501H",4,"pugilato","professionistico","maschile",19),
(null,"FNTMRC03T24H501H",6,"golf","dilettantistico","maschile",19),
(null,"PRNGCM00D24D612C",6,"golf","professionistico","maschile",22);
```

In accordo con il vincolo posto in fase di progettazione che impone che un utente possa recensire solo il centro sportivo è a disposizione il seguente trigger.

Per ogni nuova recensione effettuata da un Utente siccome se in questa non è stato segnato l'id del centro sportivo al quale è dedicata oppure è stato fatto ma in modo sbagliato ,il trigger setCS si occuperà di attribuire il valore corretto al campo idCentro.

```
DELIMITER !!
DROP TRIGGER IF EXISTS setCS;
CREATE TRIGGER setCS
BEFORE INSERT ON Recensione
FOR EACH ROW
BEGIN
DECLARE idCS INT;
SET idCS = (SELECT idCentro FROM Utente WHERE Utente.id = NEW.idUtente);
IF NEW.idCentro != idCS THEN
    SET NEW.idCentro = idCS;
END IF;
IF NEW.idCentro is null THEN
    SET NEW.idCentro = idCS;
END IF;
END !!
DELIMITER ;
```

sono stati usati due IF separati perché nel caso in cui nell'inserire la recensione a idCentro fosse attribuito null allora una sola condizione IF non sarebbe stata in grado di gestirlo. Questo perché null non è un tipo di dato e se utilizzato come termine di confronto(!= < ...) ritornerà sempre null.

```
**INSERTS SULLA TABELLA RECENSIONE**
insert into Recensione values
(null,1,null,"2020-05-09","3","3","4"),
(null,2,null,"2020-01-19","3","5","4"),
(null,4,null,"2021-03-28","5","3","4"),
(null,9,null,"2019-10-30","4","5","4"),
(null,16,null,"2021-04-06","3","2","4"),
(null,28,null,"2020-11-09","5","1","4"),
```

```
(null,30,null,"2021-04-10","3","1","2"),
(null,22,null,"2021-01-03","1","5","4"),
(null,1,null,"2020-12-14","3","3","4"),
(null,5,null,"2020-07-31","3","3","4"),
(null,1,null,"2021-09-03","3","3","4"),
(null,18,null,"2020-06-11","3","5","4"),
(null,14,null,"2021-05-23","1","3","4"),
(null,27,null,"2020-08-19","3","3","5");
-----
```

Un vincolo descritto in fase di progettazione impone che un utente non faccia più di una recensione circa ogni 5 mesi cioè ogni 150 giorni.

```
DELIMITER !!
DROP TRIGGER IF EXISTS checkRecensione;
CREATE TRIGGER checkRecensione
BEFORE INSERT ON Recensione
FOR EACH ROW
BEGIN
DECLARE msg VARCHAR(200);
DECLARE dateLast DATE DEFAULT NULL;
DECLARE days INT UNSIGNED DEFAULT NULL;
SET dateLast = (SELECT MAX(data) FROM Recensione WHERE
NEW.idUtente=Recensione.idUtente);
IF dateLast is not null THEN
SET days = (SELECT CAST(DATEDIFF(new.data,dateLast) AS UNSIGNED));
END IF;
IF (days<150 AND days is not null) THEN
SET msg = concat('ERROR.L'ULTIMA RECENSIONE RISALE A MENO DI 150 GIORNI. Data
ultima recensione:',dateLast);
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
END IF;
END !!
DELIMITER ;
```

```
/*prima insert*/
mysql> insert into Recensione values(null,27,null,"2021-06-19","4","2","5");
Query OK, 1 row affected (0.00 sec)
/*seconda insert in conflitto con la prima poiché non rispetta il vincolo*/
mysql> insert into Recensione values(null,27,null,"2021-06-30","4","2","5");
ERROR 1644 (45000): ERROR.L'ULTIMA RECENSIONE RISALE A MENO DI 150 GIORNI. Data
ultima recensione:2021-06-19
```

```
*PER RESTITUIRE LA MEDIA DI OGNI RECENSIONE:
SELECT (CAST(struttura AS UNSIGNED)+CAST(gestione AS UNSIGNED)+CAST(staffTecnico
as UNSIGNED))/3 FROM Recensione;*
```

Si impone che un iscritto non possa donare sangue più di una volta ogni 90 giorni. Pertanto non si può inserire un record nella tabella Donazione che rappresenti la donazione di un Iscritto che ha donato meno di 90 giorni prima.

```
DELIMITER !!
DROP TRIGGER IF EXISTS checkDonazione;
CREATE TRIGGER checkDonazione
BEFORE INSERT ON Donazione
FOR EACH ROW
BEGIN
DECLARE msg VARCHAR(128);
DECLARE dateLast DATE DEFAULT NULL;
DECLARE days INT UNSIGNED DEFAULT NULL;
SET dateLast = (SELECT MAX(data) FROM Donazione WHERE
NEW.codiceFiscale=Donazione.codiceFiscale);
IF dateLast is not null THEN
SET days = (SELECT CAST(DATEDIFF(new.data,dateLast) AS UNSIGNED));
END IF;
IF (days<90 AND days is not null) THEN
SET msg = concat('ERROR.L'ULTIMA DONAZIONE RISALE A MENO DI 90 GIORNI. Data
ultima donazione:',dateLast);
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
END IF;
END !!
DELIMITER ;
```

*/*prima insert*/*

```
mysql> insert into Donazione values (null,"DSNPLA75R01H501K","2020-06-20",1);
Query OK, 1 row affected (0.00 sec)
```

*/*seconda insert in conflitto con la prima*/*

```
mysql> insert into Donazione values (null,"DSNPLA75R01H501K","2020-06-29",1);
ERROR 1644 (45000): ERROR.L'ULTIMA DONAZIONE RISALE A MENO DI 90 GIORNI. Data
ultima donazione:2020-06-20
```

****INSERTS SULLA TABELLA DONAZIONE****

```
//hanno tutte esito positivo poiché conformi ai vincoli descritti
insert into Donazione values (null,"PRNGCM00D24D612C","2021-11-05",10);
insert into Donazione values (null,"PSCDGI83P09H703Z","2021-08-10",5);
insert into Donazione values (null,"RSSJPL67A30F205H","2019-12-04",8);
insert into Donazione values (null,"FRRMRC99D03D810Y","2019-11-21",2);
insert into Donazione values (null,"FRRMRC99D03D810Y","2020-06-05",2);
insert into Donazione values (null,"BRRRCC70C50G482B","2019-07-16",9);
```

VIEWS

Le "viste mysql" sono uno strumento molto utile che può essere utilizzato per vari scopi. Uno degli scopi principali è la sicurezza. Infatti se si vuole fornire ad un utente del database un determinato insieme di dati escludendone altri gli si può fornire anziché tutte le tabelle e la possibilità farvi delle query/insert/update/delete... direttamente la vista stessa.

La vista o view è infatti il risultato di una query che è stata memorizzata all'interno del database come oggetto.

Per il database che si sta realizzando le view potrebbero essere implementate per fornire un accesso ai dati da parte dell'iscritto ad uno dei centri sportivi. Questo accesso al database deve essere limitato solo ad alcune informazioni e al massimo con potere di lettura. Sarebbe disastroso se ogni iscritto potesse eliminare o aggiornare records. Per quest'ultimo motivo alle vie che verranno create in seguito sarà omessa la parte WITH CHECK OPTION. Di base infatti non si può modificare un database tramite le views.

Le seguenti views saranno utili all'iscritto solo per leggere alcune informazioni. Sono prive di "WITH CHECK OPTION" pertanto non gli sarà concessa la facoltà di modificare i contenuti delle tabelle.

L'iscritto deve poter conoscere nome e indirizzo di tutti i centri sportivi che partecipano al progetto. Per questo si utilizzerebbe una view molto semplice ma altrettanto importante.

```
select nome,  
citta,  
indirizzo,  
telefono,  
email,  
sitoWeb  
from CentroSportivo order by citta;
```

```
CREATE VIEW CENTRISPORTIVI AS  
select nome,  
citta,  
indirizzo,  
telefono,  
email,  
sitoWeb  
from CentroSportivo order by citta;
```

La seguente select è molto utile per l'iscritto poiché nel caso in cui cercasse uno specialista che si occupa di una determinata professione può farlo senza intermediari. Accede direttamente al risultato di questa select tramite una view e può individuare lo specialista che sta cercando, il centro sanitario in cui opera ed i vari contatti.

```
select CentroSanitario.nome as nomeCentro,  
CentroSanitario.citta as cittaCentro,  
CentroSanitario.indirizzo as indirizzoCentro,  
CentroSanitario.telefono as contattoCentro,  
CentroSanitario.email as emailCentro,  
CentroSanitario.sitoWeb as sitoCentro,  
professione,  
Specialista.nome as nomeSpecialista,  
Specialista.cognome as cognomeSpecialista,  
Specialista.telefono as contattoSpecialista,
```

```
Specialista.email as emailSpecialista
from CentroSanitario,Specialista
where CentroSanitario.id=idCentro
order by professione;
```

```
CREATE VIEW SPECIALISTI AS
select CentroSanitario.nome as nomeCentro,
CentroSanitario.citta as cittaCentro,
CentroSanitario.indirizzo as indirizzoCentro,
CentroSanitario.telefono as contattoCentro,
CentroSanitario.email as emailCentro,
CentroSanitario.sitoWeb as sitoCentro,
professione,
Specialista.nome as nomeSpecialista,
Specialista.cognome as cognomeSpecialista,
Specialista.telefono as contattoSpecialista,
Specialista.email as emailSpecialista
from CentroSanitario,Specialista
where CentroSanitario.id=idCentro
order by professione;
```

Si nota che esiste una tabella di nome Specialista. La view appena creata ha il nome di SPECIALISTI che è ben differente. Non si può infatti attribuire ad una view lo stesso nome di una tabella. Ciò genererebbe ambiguità.

```
select
cittaCentro,indirizzoCentro,professione,cognomeSpecialista,contattoSpecialista
from SPECIALISTI;
```

La seguente select mette a disposizione degli iscritti i nomi e le informazioni relative ai centri per la donazione del sangue.

```
select nome,
indirizzo,
citta,
telefono as contatto,
email,
sitoWeb
from CentroDonazione order by citta;
```

```
CREATE VIEW CENTRIDONAZIONE AS
select nome,
indirizzo,
citta,
telefono as contatto,
email,
sitoWeb
from CentroDonazione order by citta;
```

Nuovamente è da notare che il nome della view creata si differenzia dal nome della tabella CentroDonazione.

Inoltre osserviamo che come nella select della view precedente non sono contemplati gli id dei centri in quanto non rappresentano un'informazione interessante per gli iscritti.

La seguente select mette a disposizione degli iscritti le recensioni fatte sui centri sportivi dagli utenti che li frequentano. Inoltre all'interno della select è effettuato il calcolo della media dei tre voti per ogni recensione.

```
select CentroSportivo.nome,
data as dataRecensione,
struttura as votoStruttura,
gestione as votoGestione,
```

```

staffTecnico as votoStaffTecnico,
(CAST(struttura AS UNSIGNED)+CAST(gestione AS UNSIGNED)+CAST(staffTecnico as UNSIGNED))/3
as mediaVoti
from Recensione,CentroSportivo
where idCentro=CentroSportivo.id;

```

```

CREATE VIEW RECENSIONI AS
select CentroSportivo.nome,
data as dataRecensione,
struttura as votoStruttura,
gestione as votoGestione,
staffTecnico as votoStaffTecnico,
(CAST(struttura AS UNSIGNED)+CAST(gestione AS UNSIGNED)+CAST(staffTecnico as
UNSIGNED))/3 as mediaVoti
from Recensione,CentroSportivo
where idCentro=CentroSportivo.id;

```

Si osserva che nella view non sono mostrati né id del centro sportivo e nemmeno l'id dell'utente. Anche se l'id dell'utente non rivela l'identità dell'utente si preferisce non mostrarlo all'iscritto.

Gli iscritti hanno bisogno di conoscere quali sport sono praticati nei vari centri sportivi se indoor o outdoor o a che livello se agonistico oppure dilettantistico.

```

select CentroSportivo.nome as nome,
CentroSportivo.citta as citta,
CentroSportivo.indirizzo as indirizzo,
sport,
livello,
spazio
from CentroSportivo,Praticato
where Praticato.idCentro=CentroSportivo.id
order by sport;

```

```

CREATE VIEW SPORTPRATICATI AS
select CentroSportivo.nome as nome,
CentroSportivo.citta as citta,
CentroSportivo.indirizzo as indirizzo,
sport,
livello,
spazio
from CentroSportivo,Praticato
where Praticato.idCentro=CentroSportivo.id
order by sport;

```

Si nota che se un iscritto cerca un determinato sport è facilitato poiché i record sono disposti in ordine alfabetico rispetto ai nomi degli sport.

Un aspetto secondario che riguarda l'utilizzo delle views è legato al fatto che si potrebbe preferire memorizzare queries particolarmente complesse.

Dopo aver creato le precedenti views bisogna però occuparsi della gestione dei privilegi altrimenti queste non avrebbero senso.

GESTIONE DEI PRIVILEGI

Come spiegato nel corso della progettazione la base di dati è sia destinato ad un uso amministrativo, ma anche degli iscritti. Infatti un iscritto può effettuare recensioni oppure cercare determinate informazioni come quelle sui centri sanitari convenzionati. Bisogna dunque attuare una serie di misure volte alla gestione dei privilegi che un iscritto o un amministratore può avere. Supponiamo che ci sia un amministratore per ogni centro sportivo che partecipa al progetto "Periodonikes". Allora bisognerà creare degli USERS che possano accedere al database con quasi tutti i privilegi.

```
CREATE USER 'amministratore01'@'localhost' IDENTIFIED BY 'password';
```

```
GRANT SELECT,INSERT,UPDATE,DELETE ON periodonikes.* TO 'amministratore01'@'localhost';
```

È stato appena creato un utente che come amministratore può accedere a tutte le tabelle del database in lettura, scrittura, cancellazione e modifica.

Un iscritto invece ha solamente la possibilità di leggere da determinate views e di effettuare inserimenti sulla tabella Recensione.

```
CREATE USER 'iscritto01'@'localhost' IDENTIFIED BY 'password';
```

```
GRANT INSERT ON periodonikes.Recensione TO 'iscritto01'@'localhost';
```

```
GRANT SELECT ON periodonikes.CENTRISPORTIVI TO 'iscritto01'@'localhost';  
GRANT SELECT ON periodonikes.SPORTPRATICATI TO 'iscritto01'@'localhost';  
GRANT SELECT ON periodonikes.SPECIALISTI TO 'iscritto01'@'localhost';  
GRANT SELECT ON periodonikes.CENTRIDONAZIONE TO 'iscritto01'@'localhost';  
GRANT SELECT ON periodonikes.RECENSIONI TO 'iscritto01'@'localhost';
```

Si osserva che all'iscritto è conferito il diritto di insert sulla tabella Recensione e non sulla view RECENSIONI. Questo perché la view RECENSIONI può essere usata solo in lettura. Inoltre l'iscritto andrà ad effettuare la lettura sulla view RECENSIONI che mette in mostra solo i contenuti che gli interessano relativi alla tabella Recensione.

QUERIES

-nome,indirizzo,sitoWeb dei Centri Sportivi di Roma.

```
mysql> select nome,indirizzo as Roma,sitoWeb from CentroSportivo where  
citta="Roma";
```

nome	Roma	sitoWeb
Sport Center	via Italo 869	www.SportCenter.it
Atletigame	via dei Draghi 345	Atletigame.com
Lucen Sport	via Conte Max 123	www.lucensport.it
Appia Sport Village	via Appia Moderna 777	www.appiasport.com

4 rows in set (0.00 sec)

-nome,indirizzo e telefono dei Centri Sportivi in cui si praticano sport a livello dilettantistico e nome di questi sport.La tabella generata dovr  mostrare i record ordinati in ordine alfabetico degli sport.

```
mysql> select sport as  
sport,CentroSportivo.nome,CentroSportivo.citta,CentroSportivo.indirizzo,CentroSportivo.te  
lefono from CentroSportivo,Praticato where Praticato.livello="dilettantistico" and  
Praticato.idCentro=CentroSportivo.id order by sport;
```

sport	nome	citta	indirizzo	telefono
calcio	Lucen Sport	Roma	via Conte Max 123	088678574
golf	Lucen Sport	Roma	via Conte Max 123	088678574
nuoto	Sport Center	Roma	via Italo 869	068609452
nuoto	Sportessere	Frosinone	via Carducci 45	068609487
pallacanestro	Sportessere	Frosinone	via Carducci 45	068609487
pallanuoto	Sportessere	Frosinone	via Carducci 45	068609487
pallavolo	Sport Center	Roma	via Italo 869	068609452
pallavolo	Appia Sport Village	Roma	via Appia Moderna 777	088645328
pugilato	Sport Center	Roma	via Italo 869	068609452
pugilato	Sportessere	Frosinone	via Carducci 45	068609487
rugby	Salerno Sport	Salerno	via Vietri 56	089407123
tennis	Country Club	Napoli	via Pozzi 34	089306789

12 rows in set (0.00 sec)

-matricola,nome,cognome e sport(con livello) di Competenza dei membri del personale tecnico che hanno la laurea come titolo di studio

```
mysql> select PersonaleTecnico.matricola as  
mat,nome,cognome,Competenza.sport,Competenza.livello from  
PersonaleTecnico,Competenza where titolo="laurea" and  
Competenza.matricola=PersonaleTecnico.matricola;
```

mat	nome	cognome	sport	livello
2	Biagio	Napolitani	nuoto	professionistico
2	Biagio	Napolitani	nuoto	dilettantistico
2	Biagio	Napolitani	pallanuoto	professionistico
2	Biagio	Napolitani	pallanuoto	dilettantistico
6	Alessandra	Pinto	tennis	professionistico
8	Raffaele	Costa	calcio	dilettantistico

6 rows in set (0.00 sec)

-matricola,nome,cognome,ruolo del membro del personale amministrativo con diploma che ha un impiego nel centro sportivo "Appia Sport Village"

```
mysql> select  
matricola, PersonaleAmministrativo.nome, PersonaleAmministrativo.cognome, qualifica  
from CentroSportivo, PersonaleAmministrativo where CentroSportivo.nome="Appia  
Sport Village" and CentroSportivo.id=PersonaleAmministrativo.idCentro and  
titolo="diploma";
```

matricola	nome	cognome	qualifica
2	Flaviana	Toscani	segretario
9	Vincenzo	Petrangeli	dirigente

2 rows in set (0.00 sec)

-query fatta da un iscritto

nome,cognome,centroSanitario,indirizzoCentroSanitario,cittaCentroSanitario,telefono degli osteopati e degli ortopedici che lavorano in un centro sanitario di Roma.

```
select professione,nomeSpecialista as nome,cognomeSpecialista as  
cognome,contattoSpecialista as contatto,indirizzoCentro as Roma from SPECIALISTI  
where cittaCentro="Roma" and (professione="osteopata" or  
professione="ortopedico");
```

professione	nome	cognome	contatto	Roma
ortopedico	Danila	Linari	3451796216	via dei Mattarella 644
ortopedico	Chiara	Bizanti	3339824416	via Appia Moderna 157
osteopata	Catia	Paletta	3771796846	via Davide 865
osteopata	Ivana	De Rossi	068997846	via Davide 865

4 rows in set (0.00 sec)

si osserva che la query è fatta sulla view SPECIALISTI in quanto all'iscritto è consentito fare select solo sulla view relativa ai centri sanitari e specialisti

-media delle recensioni di ogni centro sportivo

```
mysql> select nome,avg(mediaVoti) as avgMedia from RECENSIONI group by nome;
```

nome	avgMedia
Atletigame	3.33330000
Country Club	3.33330000
Lucen Sport	2.83335000
Salerno Sport	3.16665000
Sport Center	3.55555000
Sportessere	4.00000000
Svago & Forma	4.00000000

7 rows in set (0.00 sec)

*essendo una select sulla view RECENSIONI questa query può anche essere effettuata da un iscritto

-matricola,nome,cognome dei membri del personale tecnico che sono istruttori ed hanno almeno il diploma

```
mysql> select x.matricola,x.nome,x.cognome,y.titolo from PersonaleTecnico
x,PersonaleTecnico y where x.qualifica="istruttore"and y.titolo is not null and
x.matricola=y.matricola;
```

matricola	nome	cognome	titolo
1	Gabriele	Zito	diploma
2	Biagio	Napolitani	laurea
4	Loredana	Conti	diploma
7	Fernando	Pugliesi	diploma
8	Raffaele	Costa	laurea
12	Fabrizio	Milani	diploma

6 rows in set (0.00 sec)

-nome,cognome,eta degli utenti che praticano almeno uno sport a livello professionistico

```
mysql> select distinct Iscritto.nome,Iscriotto.cognome,Utente.eta from
Utente,Iscriotto where Utente.codiceFiscale=Iscriotto.codiceFiscale and
Utente.livello="professionistico";
```

nome	cognome	eta
Sofia	Chiari	15
Paolo	De Santis	45
Lucia	Destro	19
Marco	Fanti	17
Marco	Ferri	22
Giacomo	Parini	21
Tiziana	Rampini	41
Roberta	Rossi	65
Andrea	Stankovic	29
Giovanni	Veltri	28

10 rows in set (0.00 sec)

la clausola distinct serve perché nel caso in cui un utente praticasse più di uno sport a livello agonistico avremmo nella tabella risultante due record identici

-matricola,sport,livello,centro sportivo dei membri del personale tecnico che lavorano nei centri sportivi NON della capitale

mysql> select

Impiego.matricola,CentroSportivo.nome,Competenza.sport,Competenza.livello from CentroSportivo,Impiego,Competenza where CentroSportivo.citta!="Roma" and CentroSportivo.id=Impiego.idCentro and Competenza.matricola=Impiego.matricola order by Competenza.matricola;

matricola	nome	sport	livello
1	Salerno Sport	pugilato	dilettantistico
1	Svago & Forma	pugilato	dilettantistico
1	Salerno Sport	rugby	professionistico
1	Svago & Forma	rugby	professionistico
2	Sportessere	nuoto	professionistico
2	Sportessere	pallanuoto	dilettantistico
2	Sportessere	nuoto	dilettantistico
2	Sportessere	pallanuoto	professionistico
5	Salerno Sport	rugby	dilettantistico
5	Salerno Sport	calcio	professionistico
6	Country Club	tennis	professionistico
7	Country Club	tennis	professionistico
7	Country Club	tennis	dilettantistico
9	Sportessere	pallavolo	dilettantistico
9	Sportessere	nuoto	dilettantistico
9	Sportessere	pallanuoto	professionistico
10	Svago & Forma	calcio	professionistico
10	Salerno Sport	rugby	dilettantistico
10	Svago & Forma	rugby	professionistico
10	Salerno Sport	calcio	professionistico
10	Svago & Forma	rugby	dilettantistico
10	Salerno Sport	rugby	professionistico

22 rows in set (0.00 sec)

-nome,cognome,sport,livello degli utenti di sesso femminile

mysql> select Utente.codiceFiscale,nome,cognome,sport,livello from Iscritto,Utente where Utente.sesso="femminile" and Utente.codiceFiscale=Iscritto.codiceFiscale;

codiceFiscale	nome	cognome	sport	livello
BRRRCC70C50G482B	Rebecca	Barry	pallavolo	dilettantistico
CHRSF005S52H501J	Sofia	Chiari	pallavolo	professionistico
CHRSF005S52H501J	Sofia	Chiari	pallavolo	professionistico
CHRSF005S52H501J	Sofia	Chiari	pallanuoto	professionistico
CRTLNS73C41A345V	Alessandra	Cortesi	pugilato	dilettantistico
DSTLCU02E49L719G	Lucia	Destro	calcio	professionistico
RSSRRT56C61H501C	Roberta	Rossi	tennis	professionistico
RSSRRT56C61H501C	Roberta	Rossi	nuoto	dilettantistico
STNNDR92D58H703R	Andrea	Stankovic	tennis	professionistico
STNNDR92D58H703R	Andrea	Stankovic	nuoto	professionistico
STNNDR92D58H703R	Andrea	Stankovic	tennis	professionistico

11 rows in set (0.00 sec)

-età media degli utenti di sesso maschile ed età media degli utenti di sesso femminile che frequentano i centri sportivi

-tutti gli sport praticati a livello dilettantistico e in uno spazio interno nei vari centri sportivi

```
mysql> select distinct(sport) from Praticato where spazio="indoor" and  
livello="dilettantistico" order by sport;
```

```
+-----+  
| sport      |  
+-----+  
| nuoto      |  
| pallacanestro |  
| pallanuoto |  
| pallavolo  |  
| pugilato   |  
+-----+
```

5 rows in set (0.00 sec)

-numero degli sport praticati a livello dilettantistico e in uno spazio interno nei vari centri sportivi

```
mysql> select count(nid.nomeSport) as SportIndoor from (select distinct(sport)  
as nomeSport from Praticato where Praticato.spazio="indoor" and  
Praticato.livello="dilettantistico") as nid;
```

```
+-----+  
| SportIndoor |  
+-----+  
|           5 |  
+-----+
```

1 row in set (0.00 sec)

è utilizzata la query precedente come nidificata

-media degli iscritti di sesso maschile

```
mysql> select avg(IscrittiMaschi.eta) as etaMedia from (select  
distinct(codiceFiscale),eta from Utente where sesso="maschile") as  
IscrittiMaschi;
```

```
+-----+  
| etaMedia |  
+-----+  
| 39.3636 |  
+-----+
```

1 row in set (0.00 sec)

la query si sarebbe potuta fare direttamente sulla tabella Iscritto ma per renderla più complessa è stata fatta sulla tabella utenti. Gli iscritti sono individuati tramite distinct(codiceFiscale). Infatti un iscritto potendo rappresentare più utenti può avere più id, ma il codiceFiscale rimarrà ovviamente unico

-età media degli utenti che frequentano un centro sportivo

```
mysql> select nome as nomeCentro,avg(eta) as etaMedia from CentroSportivo,Utente
where Utente.idCentro=CentroSportivo.id group by CentroSportivo.nome;
```

nomeCentro	etaMedia
Appia Sport Village	43.0000
Atletigame	35.0000
Country Club	45.8000
Lucen Sport	34.8000
Salerno Sport	27.2000
Sport Center	43.7500
Sportessere	24.3333
Svago & Forma	42.0000

8 rows in set (0.00 sec)

-nome,cognome,telefono degli iscritti che hanno effettuato almeno una donazione

dopo la data 01/05/2019

```
mysql> select Iscritto.nome,Iscritto.cognome,Iscritto.telefono from Iscritto,
(select distinct(Donazione.codiceFiscale) as codFiscale from Donazione where
data>"2019-01-05") as Donazioni where
Donazioni.codFiscale=Iscritto.codiceFiscale;
```

nome	cognome	telefono
Rebecca	Barry	06867478
Paolo	De Santis	06869526
Marco	Ferri	06859581
Giacomo	Parini	06869771
Diego	Pascoli	06959176
Jean Paul	Rossi	06869521

6 rows in set (0.01 sec)

-nome,indirizzo,città,mediaVoti dei centri sportivi NON di Roma che hanno una media dei voti superiore a 3.2

```
mysql> select
CENTRISPORTIVI.nome,CENTRISPORTIVI.indirizzo,CENTRISPORTIVI.citta,Temp.punteggio
Recensioni from(select nome,avg(mediaVoti) as punteggioRecensioni from
RECENSIONI group by nome) as Temp,CENTRISPORTIVI where
Temp.nome=CENTRISPORTIVI.nome and punteggioRecensioni>3.2 and
CENTRISPORTIVI.citta!="Roma";
```

nome	indirizzo	citta	punteggioRecensioni
Sportessere	via Carducci 45	Frosinone	4.00000000
Svago & Forma	piazza Repubblica 78	Velletri	4.00000000
Country Club	via Pozzi 34	Napoli	3.33330000

3 rows in set (0.01 sec)

-nome ed id dei centri sportivi e dei centri donazione con i quali sono convenzionati e la data a cui risale l'inizio della convenzione

```
mysql> select x.idCentroSportivo as idCS,y.nome as CentroSportivo,x.idCentroDonazione as idCD,z.nome as CentroDonazione,x.data as dataConvenzione from Convenzione_CD x,CentroSportivo y,CentroDonazione z where x.idCentroSportivo=y.id and z.id=x.idCentroDonazione;
```

idCS	CentroSportivo	idCD	CentroDonazione	dataConvenzione
8	Appia Sport Village	1	AVIS	2021-05-12
2	Sportessere	2	Centro Braschi	2020-05-27
3	Svago & Forma	2	Centro Braschi	2020-11-03
3	Svago & Forma	3	Associazione Donatori	2019-08-05
1	Sport Center	4	AVIS	2020-08-09
5	Country Club	5	Centro EMA	2021-03-20
6	Salerno Sport	5	Centro EMA	2020-10-28
6	Salerno Sport	6	AVIS	2021-01-15
1	Sport Center	7	Centro Nazionale Donazioni	2019-12-01
2	Sportessere	7	Centro Nazionale Donazioni	2019-04-30
4	Atletigame	8	AVIS	2019-10-18
8	Appia Sport Village	9	NULL	2019-11-16
1	Sport Center	10	Progetto EMAS	2020-11-26

13 rows in set (0.00 sec)

-nome ed id dei centri sportivi e dei centri sanitari con i quali sono convenzionati e la data a cui risale l'inizio della convenzione

```
mysql> select x.idCentroSportivo as idCSport,y.nome as CentroSportivo,x.idCentroSanitario as idCSan,z.nome as CentroSanitario,x.data as dataConvenzione from Convenzione_CS x,CentroSportivo y,CentroSanitario z where x.idCentroSportivo=y.id and z.id=x.idCentroSanitario;
```

idCSport	CentroSportivo	idCSan	CentroSanitario	dataConvenzione
4	Atletigame	1	C&V	2021-03-18
8	Appia Sport Village	1	C&V	2020-10-30
2	Sportessere	2	Centro Pugliese	2021-05-07
3	Svago & Forma	3	Fisiosports	2020-12-01
1	Sport Center	4	Polispecialistica Rossi	2020-04-23
8	Appia Sport Village	4	Polispecialistica Rossi	2019-11-03
5	Country Club	5	Studio Corpore Sano	2021-02-05
6	Salerno Sport	5	Studio Corpore Sano	2020-09-21
6	Salerno Sport	6	Studio Paideia	2020-04-22
1	Sport Center	7	Almamed	2019-06-03
3	Svago & Forma	7	Almamed	2019-07-29
1	Sport Center	8	Poliambulatorio Cecconi	2020-08-24
8	Appia Sport Village	8	Poliambulatorio Cecconi	2021-01-10
4	Atletigame	9	NULL	2020-05-08
1	Sport Center	10	Progetto Salute srl	2020-08-16

15 rows in set (0.00 sec)

-nome,cognome,data di nascita,email di 5 iscritti di sesso femminile

```
mysql> select nome,cognome,dataNascita,email from Iscritto where sesso="femminile" limit 5;
```

nome	cognome	dataNascita	email
Rebecca	Barry	1970-03-10	rebeccabarry@gmail.com
Sofia	Chiari	2005-11-12	sofiachiari@hotmail.it
Alessandra	Cortesi	1973-03-01	alessandracortesi@libero.it
Lucia	Destro	2002-05-09	luciadestro@libero.it
Roberta	Rossi	1956-03-21	robertarossi@gmail.it

5 rows in set (0.00 sec)

-data visita,(nome,cognome e codiceFiscale del visitato) e professione del visitante

```
mysql> select Iscritto.codiceFiscale,Iscritto.nome,Iscritto.cognome,Visita.data as dataVisita,Specialista.professione as professioneVisitante from Visita,Iscritto,Specialista where Visita.codiceFiscale=Iscritto.codiceFiscale and Visita.matricolaSpecialista=Specialista.matricola;
```

codiceFiscale	nome	cognome	dataVisita	professioneVisitante
RSSRRT56C61H501C	Roberta	Rossi	2020-11-19	osteopata
PRNGCM00D24D612C	Giacomo	Parini	2021-05-16	neurologo
RSSJPL67A30F205H	Jean Paul	Rossi	2019-04-04	ortopedico
PSCDGI83P09H703Z	Diego	Pascoli	2021-10-09	fisioterapista
CRTLSN73C41A345V	Alessandra	Cortesi	2020-02-22	fisioterapista
RSSRRT56C61H501C	Roberta	Rossi	2021-01-31	oculista
BRRRC70C50G482B	Rebecca	Barry	2019-06-18	osteopata
FNTMRC03T24H501H	Marco	Fanti	2020-09-26	medico di base
STNND92D58H703R	Andrea	Stankovic	2020-12-05	otorino
STNND92D58H703R	Andrea	Stankovic	2019-11-21	oculista

10 rows in set (0.00 sec)

-media dei voti presenti nelle recensioni per ogni centro sportivo con il nome del centro sportivo

```
mysql> select nome,truncate(tots/numR,2) as votoStruttura,truncate(totG/numR,2) as votoGestione,truncate(totST/numR,2) as votoStaffTecnico from (select count(idCentro) as numR,idCentro,sum(struttura) as totS,sum(gestione) as totG,sum(staffTecnico) as totST from Recensione group by idCentro) as X,CentroSportivo where idCentro=CentroSportivo.id group by idCentro;
```

nome	votoStruttura	votoGestione	votoStaffTecnico
Sport Center	3.16	3.16	4.33
Sportessere	5.00	3.00	4.00
Svago & Forma	3.00	5.00	4.00
Aletigame	4.00	2.00	4.00
Country Club	1.00	5.00	4.00
Salerno Sport	3.50	3.00	3.00
Lucen Sport	2.00	2.50	4.00

7 rows in set (0.00 sec)

il risultato di questa query verrà usato per la realizzazione di una collection mongoDB

//ALTRE QUERIES SONO MOSTRATE IMPLICITAMENTE NELLA CREAZIONE DEI TRIGGERS

QUERIES IN ALGEBRA RELAZIONALE

-matricola,nome,cognome,centro sportivo dei membri del personale amministrativo che lavorano nei centri sportivi NON della capitale

Π (matricola,nome,cognome,nomeCentro)

(PersonaleAmministrativo \mid X \mid ρ (CentroSportivo))
(idCentro=id) (nomeCentro<-nome)

-

(σ ((PersonaleAmministrativo \mid X \mid ρ (CentroSportivo))))
(citta="Roma") (idCentro=id) (nomeCentro<-nome)
)

il "-" al centro è l'operatore di differenza

-nome,indirizzo e telefono dei Centri Sportivi in cui si praticano sport a livello dilettantistico e nome di questi sport

Π (sport,nome,indirizzo,telefono)

(
 σ (Praticato \mid X \mid CentroSportivo)
(livello="dilettantistico) (idCentro=id)
)

MongoDB

Per la creazione e il riempimento delle collections mongoDB che alludono alle tabelle MYSQL si è scelto di estrapolarne tramite select i record in formato csv.

Una volta ottenuto il formato csv di una tabella questo è stato trasformato in formato JSON utilizzando un convertitore.

Tabella CentroSportivo

-il risultato della select è memorizzato in un file "CentroSportivo.csv"

```
mysql> SELECT * FROM CentroSportivo INTO OUTFILE
'/var/lib/mysql-files/CentroSportivo.csv' FIELDS TERMINATED BY ',' ENCLOSED BY
'' LINES TERMINATED BY '\n';
```

-Il file CentroSportivo.csv è convertito in formato json(in python quanto mostrato sotto non è altro che una lista di dizionari)

```
[
  {
    "id": 1,
    "nome": "Sport Center",
    "indirizzo": "via Italo 869",
    "citta": "Roma",
    "telefono": "068609452",
    "email": "SportCenter@hotmail.com",
    "sito": "www.SportCenter.it"
  },
  {
    "id": 2,
    "nome": "Sportessere",
    "indirizzo": "via Carducci 45",
    "citta": "Frosinone",
    "telefono": "068609487",
    "email": "Sportessere@tin.com",
    "sito": "www.Sportessere.it"
  },
  {
    "id": 3,
    "nome": "Svago & Forma",
    "indirizzo": "piazza Repubblica 78",
    "citta": "Velletri",
    "telefono": "0894609472",
    "email": "Svago.Forma@gmail.com",
    "sito": "SvagoForma.it"
  },
  ...
]
```

Viene fatta la stessa identica cosa per la tabella "Iscritto".

Dunque i record delle tabelle sono stati mutati in oggetto json.

A seguire è mostrato l'utilizzo della libreria di python pymongo, la quale consente di effettuare operazioni in mongoDB, dalle più basilari alle più complesse.

La collection da creare è strutturata nel seguente modo:

Iscritto{

```
"Codice Fiscale":codiceFiscale,

  "dati iscritto":{
    "nome":nome,
    "cognome":cognome,
    "sesso":sesso,
    "data di nascita":data di Nascita,
    "telefono":telefono,
    "email":email,
    "indirizzo":indirizzo,
    "citta":citta,
    "codice certificato":codice certificato
  },

  "CentroSportivo":{
    "idCentro":id Centro,
    "nomeCentro":nome Centro
  }
}
```

La collection "Iscritto" è caratterizzata da avere campi che identificano l'iscritto, contengono sue informazioni personali e riferiscono a quale centro sportivo è iscritto.

In particolare queste ultime due categorie di dati sono riunite in due documenti "dati iscritto" e "CentroSportivo". Ogni iscritto infatti avrà il proprio documento "dati iscritto" e "CentroSportivo"

Per realizzare questa collection occorre operare sugli oggetti JSON ricavati precedentemente da MYSQL.

Lo script utilizzato è stato il seguente:

```
import pymongo
import json

# Connessione a MongoDB
connessione = pymongo.MongoClient("mongodb://localhost:27017/")
# Creazione del database
Periodonikes = connessione["periodonikes"]
# Creazione della collezione Iscritto
coll_iscritto=Periodonikes["Iscritto"]

#Apertura dei file JSON
f_CS = open('CentroSportivo.json')
f_I= open('Iscritto.json')
#centriSportivi e iscritti sono oggetti JSON
centriSportivi = json.load(f_CS)
```

```
iscritti = json.load(f_l)
```

```
#creazione della collezione "Iscritto"
coll_iscritto=Periodonikes["Iscritto"]
for i in range(len(iscritti)):
#inserimento del valore con campo codiceFiscale
coll_iscritto.insert_one({"Codice Fiscale":iscritti[i]["codiceFiscale"]})
#viene rilevato il codiceFiscale per poi effettuare l'update
codF=iscritti[i]["codiceFiscale"]
#creazione del documento "dati iscritto" per l'iscritto corrente
DATI = { "$set":{"dati iscritto":{"nome":iscritti[i]["nome"],"cognome":iscritti[i]
["cognome"],
" Sesso":iscritti[i][" Sesso"],"data di Nascita":iscritti[i]["dataNascita"],
"telefono":iscritti[i]["telefono"],"email":iscritti[i]["email"],
"indirizzo":iscritti[i]["indirizzo"],"citta":iscritti[i]["citta"],
"codice certificato":iscritti[i]["codiceCertificato"]}}}
#inserimento dei dati personali dell'iscritto
coll.update_one({"Codice Fiscale":codF},DATI)
#viene individuato nell'oggetto JSON "centriSportivi" il centro sportivo cui è iscritto
l'individuo corrente
for j in range(len(centriSportivi)):
if(centriSportivi[j]["id"]==iscritti[i]["idCentro"]):
#creazione del documento "Centro Sportivo" per l'iscritto corrente
CS = { "$set":{"CentroSportivo":{"idCentro":centriSportivi[j]["id"],"nomeCentro":
centriSportivi[j]["nome"]} }}
#inserimento del centro sportivo dell'iscritto
coll.update_one({"Codice Fiscale":codF},CS)
```

Collezione creata:

```
> use periodonikes
switched to db periodonikes
> show collections
Iscritto
> db.Iscritto.find()
```

```
{ "_id" : ObjectId("60e0609dea10723c95c677a7"), "Codice Fiscale" :
"BRRRCC70C50G482B", "dati iscritto" : { "nome" : "Rebecca", "cognome" : "Barry",
" Sesso" : "femminile", "data di Nascita" : "1970-03-10", "telefono" :
"06867478", "email" : "rebeccabarry@gmail.com", "indirizzo" : "via Regina
Elisabetta 12", "citta" : "Roma", "codice certificato" : 134524 },
"CentroSportivo" : { "idCentro" : 7, "nomeCentro" : "Lucen Sport" } }
```

```
{ "_id" : ObjectId("60e0609dea10723c95c677a8"), "Codice Fiscale" :
"CHRSF005S52H501J", "dati iscritto" : { "nome" : "Sofia", "cognome" : "Chiari",
" Sesso" : "femminile", "data di Nascita" : "2005-11-12", "telefono" :
"06863521", "email" : "sofiachiari@hotmail.it", "indirizzo" : "via Italo 560",
"citta" : "Roma", "codice certificato" : 98524 }, "CentroSportivo" :
{ "idCentro" : 1, "nomeCentro" : "Sport Center" } }
```

```
{ "_id" : ObjectId("60e0609dea10723c95c677a9"), "Codice Fiscale" :
"CRTLSN73C41A345V", "dati iscritto" : { "nome" : "Alessandra", "cognome" :
"Cortesi", " Sesso" : "femminile", "data di Nascita" : "1973-03-01", "telefono" :
"06860023", "email" : "alessandracortesi@libero.it", "indirizzo" : "via
dell'Angelo 45", "citta" : "Roma", "codice certificato" : 54324 },
```

```
"CentroSportivo" : { "idCentro" : 7, "nomeCentro" : "Lucen Sport" } },
.....{...}
```

QUERY BANALE PER VEDERE SE I DATI DI MYSQL E MONGO COINCIDONO

-codice fiscale,nome,cognome,data di nascita e nome del centro sportivo degli iscritti di sesso femminile

```
> db.Iscritto.find({"dati iscritto.sesso":"femminile"}, {"Codice Fiscale":1, "dati
iscritto.nome":1, "dati iscritto.cognome":1, "dati iscritto.data di
Nascita":1, "CentroSportivo.nomeCentro":1})
{ "_id" : ObjectId("60e0609dea10723c95c677a7"), "Codice Fiscale" :
"BRRRCC70C50G482B", "dati iscritto" : { "nome" : "Rebecca", "cognome" : "Barry",
"data di Nascita" : "1970-03-10" }, "CentroSportivo" : { "nomeCentro" : "Lucen
Sport" } }
{ "_id" : ObjectId("60e0609dea10723c95c677a8"), "Codice Fiscale" :
"CHRSF005S52H501J", "dati iscritto" : { "nome" : "Sofia", "cognome" : "Chiari",
"data di Nascita" : "2005-11-12" }, "CentroSportivo" : { "nomeCentro" : "Sport
Center" } }
{ "_id" : ObjectId("60e0609dea10723c95c677a9"), "Codice Fiscale" :
"CRTLSN73C41A345V", "dati iscritto" : { "nome" : "Alessandra", "cognome" :
"Cortesi", "data di Nascita" : "1973-03-01" }, "CentroSportivo" : { "nomeCentro"
: "Lucen Sport" } }
{ "_id" : ObjectId("60e0609dea10723c95c677ab"), "Codice Fiscale" :
"DSTLCU02E49L719G", "dati iscritto" : { "nome" : "Lucia", "cognome" : "Destro",
"data di Nascita" : "2002-05-09" }, "CentroSportivo" : { "nomeCentro" : "Svago &
Forma" } }
{ "_id" : ObjectId("60e0609dea10723c95c677b3"), "Codice Fiscale" :
"RSSRRT56C61H501C", "dati iscritto" : { "nome" : "Roberta", "cognome" : "Rossi",
"data di Nascita" : "1956-03-21" }, "CentroSportivo" : { "nomeCentro" : "Sport
Center" } }
{ "_id" : ObjectId("60e0609dea10723c95c677b4"), "Codice Fiscale" :
"STNNDR92D58H703R", "dati iscritto" : { "nome" : "Andrea", "cognome" :
"Stankovic", "data di Nascita" : "1992-04-18" }, "CentroSportivo" :
{ "nomeCentro" : "Salerno Sport" } }
```

```
mysql> select
Iscritto.codiceFiscale,Iscritto.nome,Iscritto.cognome,Iscritto.dataNascita,Centr
oSportivo.nome as nomeCentro from Iscritto,CentroSportivo where
Iscritto.idCentro=CentroSportivo.id and Iscritto.sesso="femminile";
+-----+-----+-----+-----+-----+
| codiceFiscale | nome | cognome | dataNascita | nomeCentro |
+-----+-----+-----+-----+-----+
| BRRRCC70C50G482B | Rebecca | Barry | 1970-03-10 | Lucen Sport |
| CHRSF005S52H501J | Sofia | Chiari | 2005-11-12 | Sport Center |
| CRTLSN73C41A345V | Alessandra | Cortesi | 1973-03-01 | Lucen Sport |
| DSTLCU02E49L719G | Lucia | Destro | 2002-05-09 | Svago & Forma |
| RSSRRT56C61H501C | Roberta | Rossi | 1956-03-21 | Sport Center |
| STNNDR92D58H703R | Andrea | Stankovic | 1992-04-18 | Salerno Sport |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Effettivamente i dati ritornati coincidono...

In seguito è stata inserita nel database mongoDB di nome "periodonikes" una seconda collection che raccoglie informazioni sui centri sportivi e sulle recensioni fatte su questi.

La collection di nome "CentroSportivo" è strutturata nel seguente modo:

```
CentroSportivo{
  "id":id
  "nome":nome
    "indirizzo":{
      "via/piazza":nome,
      "citta":cognome
    },

    "contatti":{
      "telefono":telefono,
      "email":email
      "sitoWeb":sitoWeb
    }
    "recensioni":{
      "votoStruttura":votoStruttura,
      "votoGestione":votoGestione,
      "votoStaffTecnico":votoStaffTecnico
    }
}
```

Lo script eseguito è il seguente:

```
import pymongo
import json

# Connessione a MongoDB
connessione = pymongo.MongoClient("mongodb://localhost:27017/")
# Creazione del database
Periodonikes = connessione["periodonikes"]

#Apertura dei file JSON
f_CS = open('CentroSportivo.json')
f_R = open('MediaRecensioni.json')
#centriSportivi e iscritti sono oggetti JSON
centriSportivi = json.load(f_CS)
recensioni = json.load(f_R)
#creazione/apertura collection
coll_centro=Periodonikes["CentroSportivo"]

for i in range(len(centriSportivi)):
  #inserimento dei valori dei campi nome e id
  coll_centro.insert_one({"id":centriSportivi[i]["id"],"nome":centriSportivi[i]["nome"]})
  #viene rilevato l'id del centro per poi fare l'update
  idCentro=centriSportivi[i]["id"]
  #creazione del documento "indirizzo" per l'iesimo centro sportivo
  INDIRIZZO = { "$set":{"indirizzo":{"piazza/via":centriSportivi[i]["indirizzo"],
  "citta":centriSportivi[i]["citta"]}}}
```

```

CONTATTI = { "$set": { "contatti": { "telefono": centriSportivi[i][ "telefono" ],
"email": centriSportivi[i][ "email" ], "sitoWeb": centriSportivi[i][ "sito" ] } }
}
#inserimento dei documenti INDIRIZZO e CONTATTI
coll_centro.update_one( { "id": idCentro }, INDIRIZZO )
coll_centro.update_one( { "id": idCentro }, CONTATTI )
for j in range( len( recensioni ) ):
if( recensioni[j][ "idCentro" ] == idCentro ):
RECENSIONI = { "$set": { "recensioni": {
"votoStruttura": float( recensioni[j][ "votoStruttura" ] ),
"votoGestione": float( recensioni[j][ "votoGestione" ] ),
"votoStaffTecnico": float( recensioni[j][ "VotoStaffTecnico" ] ) } } }
}
#inserimento delle medie dei voti rilevati dalle recensioni
coll_centro.update_one( { "id": idCentro }, RECENSIONI )

```

Il file json "MediaRecensioni.json" è stato ricavato convertendo il file in formato csv ricavato dal comando mysql:

```

mysql> select X.idCentro, truncate( tots/numR, 2 ) as
votoStruttura, truncate( totG/numR, 2 ) as votoGestione, truncate( totST/numR, 2 )
as votoStaffTecnico from ( select count( idCentro ) as
numR, idCentro, sum( struttura ) as totS, sum( gestione ) as totG, sum( staffTecnico )
as totST from Recensione group by idCentro ) as X, CentroSportivo where
idCentro=CentroSportivo.id group by idCentro INTO OUTFILE '/var/lib/mysql-
files/MediaRecensioni.csv' FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES
TERMINATED BY '\n';
Query OK, 7 rows affected (0.00 sec)

```

la query presente all'interno del comando è una lieve modifica di una query mostrata nella parte relativa a MYSQL

Confrontiamo i risultati di una semplice query in mysql e mongoDB
-nome,indirizzo,telefono e email dei centri sportivi di Roma

MONGODB

```
> db.CentroSportivo.find({"indirizzo.citta":"Roma"},
{"nome":1,"indirizzo.piazza/
via":1,"contatti.telefono":1,"contatti.email":1,"_id":0})
{ "nome" : "Sport Center", "indirizzo" : { "piazza/via" : "via Italo 869"
}, "contatti" : { "telefono" : "068609452", "email" :
"SportCenter@hotmail.com" } }
{ "nome" : "Atletigame", "indirizzo" : { "piazza/via" : "via dei Draghi
345" }, "contatti" : { "telefono" : "088609463", "email" :
"Atletigame@libero.it" } }
{ "nome" : "Lucen Sport", "indirizzo" : { "piazza/via" : "via Conte Max
123" }, "contatti" : { "telefono" : "088678574", "email" :
"lucensport@hotmail.com" } }
{ "nome" : "Appia Sport Village", "indirizzo" : { "piazza/via" : "via
Appia Moderna 777" }, "contatti" : { "telefono" : "088645328", "email" :
"appiasport.village@gmail.it" } }
```

MYSQL

```
mysql> select nome,indirizzo,telefono,email from CentroSportivo where
citta="Roma";
```

nome	indirizzo	telefono	email
Sport Center	via Italo 869	068609452	SportCenter@hotmail.com
Atletigame	via dei Draghi 345	088609463	Atletigame@libero.it
Lucen Sport	via Conte Max 123	088678574	lucensport@hotmail.com
Appia Sport Village	via Appia Moderna 777	088645328	appiasport.village@gmail.it

4 rows in set (0.01 sec)

I risultati coincidono...

Per confrontare i tempi di risposta di MYSQL rispetto a quelli di MongoDB è stata riempita la tabella Iscritto di 100 000 record casuali. Eseguendo uno script molto semplice in python è stato creato un file contenente questi 100 000 record(casuali quindi al posto di nomi,codiciFiscali etc.. stringhe casuali) ed in formato CSV è stato dato in pasto all'istruzione "LOAD DATA INFILE..." .

In seguito i risultati di quanto fatto:

```
mysql> load data infile '/var/lib/mysql-files/"records.csv"' into table Iscritto
fields terminated by ',' enclosed by '"' lines terminated by "\n"
(codiceFiscale,idCentro,nome,cognome,sexo,dataNascita,indirizzo,citta,telefono,
email,codiceCertificato);
Query OK, 100000 rows affected (1.99 sec)
Records: 100000 Deleted: 0 Skipped: 0 Warnings: 0
```

```
mysql> select codiceFiscale,idCentro,nome,cognome,dataNascita from Iscritto;
+-----+-----+-----+-----+-----+
| codiceFiscale | idCentro | nome | cognome | dataNascita |
+-----+-----+-----+-----+-----+
| 001YTLZHSy9N7zB | 2 | nx4tjLfAt7N795C | 8tCKoGT5jhU7ZvdqWms5nbpA | 2020-07-22 |
| 003y12Zn124WbPu | 3 | q_WZNfve_r_mJDa | Pw1wcd1QT3cRo1jF6tjRjv4Yv | 2021-12-10 |
| 00BzGXNTT_eMLrG | 7 | 8ksvZTBd1GtZAet | K_hSVNIwzd3c0BSZd2TqEjai2 | 2018-04-02 |
| 00C0CBItP2RdX6o | 6 | 0J87RH9qqRYD5FH | 4BPwmB0Wng0Zha90jTFJ0lfcP | 2021-06-02 |
| 00c003EMYGnCS2j | 1 | IFUrkyi9u0AxxRH | ypxtvh4ybFKa8pQjtync4DqCs | 2020-03-21 |
| 00EndFrKqqK_4ms | 6 | 9vvuK30n8z0BDwL | v7FyoL_ZQuFcC021ISARXdQpf | 2018-06-05 |
| 00ET6aKQgAVafH0 | 2 | SUItBz9GQqJ6u1m | DZqD43UHUG6yBcDxd8XyU4zDh | 2019-07-26 |
...

```

```
...
| _ZXTpLuIsUoMchF | 8 | ypsoCDmFlmPAYM8 | zMMy1HJ9FmfPMQA5XGVvMhPiB | 2021-08-10 |
| _ZXVLz2SgOIQ8UW | 1 | X4q1g_7woYX2spL | GhJIT7rurXycywwUw0ivYgGUC | 2019-03-12 |
| _zXX4pAFavEceoi | 7 | GvRUFVW6fqY0IM | g_Z5MUCJT9phpgvj2a2yqaiqG | 2019-10-13 |
| _zyjkkhs_6wGfS | 7 | CzFhh7LKH4Ze5W2 | jHyvxpXJIQ_bdjSPFvZWZRKK3 | 2019-03-22 |
| _ZYM3kuhtD3tOfH | 1 | t0Lkw7Ycq1afwB3 | 7he9QhQIQTSFTPBUlfeTXfvh6 | 2020-09-05 |
| _ZZbvoXwFguWIYN | 3 | GYZLjMuClqK4yi5 | tIheSzme_c4KfrPb7ik1zCByR | 2021-05-23 |
| _Z_3a5Ge3QBfJ05 | 4 | euZygwsnzNbDyWe | GPiUXsqZ_OpG_K7Uv1bR10mhs | 2021-05-13 |
| _0G0LnfBnty77H | 6 | xhR5RBwoaTP9a0L | cvVtQrRB6doFXMywn4cGIgBsJ | 2019-10-16 |
| _0WU7sKUXuYfiQ | 2 | YLkSG_H6hnr1sv1 | Mcoi1cCmwmAHlpkVAMQ8774q8 | 2021-05-06 |
| _1gbk0QMvJ4ihD | 4 | douYIREouxhMLor | utdexYzhCW66s0tgDLNvtkPBc | 2018-08-09 |
| _79vEJhw2Z0wrQ | 8 | oyf27ohT9j9IT0h | Ca9hK0KWnpeq0pv88TekqVB9N | 2019-08-03 |
| _7PJKNxFam2IJR | 3 | 1SdzjQ7cQP8rznf | 7mulb05Pf2Yai_QVwsP_SQpZL | 2021-10-01 |
| _aJt_Rf2PigmZJ | 6 | s9qVxtdg2ycxY5 | AsvqnmxxJh40d7fbYysceA67u | 2019-07-13 |
| _AT8XKwfwNLZkz | 7 | c4dgJJKCWIIY30WE | 7XxiPmmnmTz1hGEmmCWIXK3nx | 2020-08-21 |
| _bLSjGZHFMA1G0 | 3 | Yr2XZM5UGhXSmKf | 40CtIVm9vPlPs0XyHkTFEAnHL | 2021-10-08 |
| _EOBa33hrJtmkY | 2 | 0EBYLcswQpuejTm | 8r3N9RxYIdwd9vK6EZucsNEbi | 2019-08-26 |
| _F_3LX0wbPXkmX | 5 | F9mGvQ5bayKetoc | DP7XtdTZ4S0Lxvbxv4XVTxn4j | 2019-02-07 |
| _GbW7XHf9NgPPi | 3 | b6mfGWzcBriUCJ_ | y5LG6SjlmQ9dIQjnFD7BVR03J | 2018-03-01 |
| _h3Vp_VZaFZ0E0 | 8 | b5QoHvYxcbe23e | Dq4TnsLEjWdAWNzAxhfAbWiG1 | 2019-06-01 |
| _iG98QBS69wiQS | 1 | wQg0uUM2HJfJrcJ | KtBUS0MMurCAWDcy8EWxA582R | 2019-06-01 |
| _ktavBIqnxKn31 | 4 | 5wc6twu9Qh701Ca | gxmJpfcOe0Il_u_csf4568SLT | 2021-09-08 |
| _kvQD6xAeWwyyuo | 5 | 3l03bmGNxrpmbWB | eLMj0Jv0JWLNyTyKnZWlxaLq | 2021-02-18 |
| _Lzv0Q8sDqguLd | 1 | 2557tlms2a_pY2_ | 4tu6dyfmdYH114gzwmp2Jakgd | 2018-04-24 |
| _n57Zz6WdJ73DF | 3 | 3lS60Hvjc6hYrs1 | saSjMuCIR8N0kbxssGLP2e6G3 | 2020-06-22 |
| _nHKh2rAiSC5bn | 3 | fjzXVy4GXZwilBC | fxE6GpAva7majjX2SFpVBwbTp | 2021-09-15 |
| _q7jJM4tCnSJ3j | 5 | 19HEoC0nKSIHE80 | i_W9L1r0Ih47UvNIhRhQkwQ0z | 2018-11-06 |
| _r9Jn1jU7kuIL1 | 3 | 0vIx62LRDHbzgvV | eQ053hwLsGm0UbQy60V5FE9aS | 2018-09-14 |
| _RvE1PWFHXsv9K | 5 | 2AWoYymG02uK9ku | 025Wp4iRFvfUKNyqCMY9qSZfQ | 2021-03-10 |
| _ryPiJUH16leb9 | 8 | Qa2po5zU_Q4gh5D | 7MOAf9i2jHoWN9WjySRNnTmVc | 2021-02-11 |
| _unQMjVz4QbCbK | 6 | AZnQ8jrwHwfwy3 | Ns18BXgKHZGdAAYyG6oqDnYw3K | 2021-02-10 |
| _XUVG4qPy1j2XK | 5 | ArKYhtIR5c90LtN | onV0LTTCRLRuX1ZJM_z9tf97aK | 2021-09-22 |
| _Yb3SzYBlx0t3w | 3 | kHibVzNkzGLpajJ | UDKQblCyzUCBWRP8JBjgrnmy8 | 2021-04-03 |
| _CITuTJknK6Mc | 1 | Zw8DAMgI7EjQYNf | 4XxSdIOHcBg78ACovcKakdPmA | 2019-12-11 |
...

```

100017 rows in set (0.08 sec)

Verrà creata una nuova collezione utilizzando lo script python che a partire dai file JSON struttura la mongoDB collection distribuendo i valori dei campi dei record in maniera conforme.
Lo script eseguito è il medesimo ma la collezione creata è differente. Così come il file JSON dei record di Iscritto (che in questo caso contiene circa 100k record) è differente.
Si è preferito creare 2 collections simili nei campi e differenti nei records perché quella già creata contiene informazioni più realistiche. Invece quella mostrata in seguito come già accennato contiene valori casuali e fittizi e la sua utilità è solo confrontare i tempi di risposta dei due DBMS.

```
> use periodonikes
switched to db periodonikes
> show collections
Iscritto
Iscritto2
> db.Iscritto2.count()
100017 //il numero di record della nuova tabella "Iscritto2"
> db.Iscritto2.find();
{ "_id" : ObjectId("60e30d60fa545a7688f43955"), "Codice Fiscale" : "001YTLZHsy9N7zB", "dati
iscritto" : { "indirizzo" :
"zjZ5MYdX8Qy0OG0tzOMMMrmd0SD1J3icFZqzCgW6YIMNdovNNFJoTIFkxWuCYQFJAk4PXcjd3728
gkXpBFvGucGT6q", "codice certificato" : "77589", "citta" : "l_AfKpXDFDep30p", "nome" :
"nx4tjLfaT7N795C", "sesso" : "femminile", "cognome" : "8tCKoGTe5jhU7ZvdqwMs5nbpa",
"telefono" : "CzukLvETVA", "data di Nascita" : "2020-07-22", "email" :
"ECkgakYsdX37BZF8ANfV9NaswuT9hHHQxZHdh_5FAy9mV" }, "CentroSportivo" : { "idCentro" :
2, "nomeCentro" : "Sportessere" } }
{ "_id" : ObjectId("60e30d60fa545a7688f43956"), "Codice Fiscale" : "003y12Zn124WbPu",
"dati iscritto" : { "indirizzo" :
"saNPcyAZG0GuPukX_81xipnVKALHvFYs02DCubNANvqJQG0cldFrFdjGYe2cZrjBA5tYX5M7E0EAp
ufMGIGFmOkugT", "codice certificato" : "71401", "citta" : "7toFiwYpz_JchEC", "nome" :
"q_WZNfve_r_mjDa", "sesso" : "femminile", "cognome" : "Pw1wcd1QT3cRo1jF6tjRjv4Yv",
"telefono" : "Pp5NIXHNyc", "data di Nascita" : "2021-12-10", "email" :
"E6aaZPNAvgwz0EowqPM2GFfbxG2fvRExscNrDzobUne1N" }, "CentroSportivo" : { "idCentro" :
3, "nomeCentro" : "Svago & Forma" } }
{ "_id" : ObjectId("60e30d60fa545a7688f43957"), "Codice Fiscale" : "00BzGXNTT_eMLrG", "dati
iscritto" : { "indirizzo" :
"Z8p0wp0PK523fdxv1dF7AUrF0Fi_t1cw55Uy2P2h6ulwXQoUNBxK2VBgXga5mbujcQXXRijlYxQP
wK75YZrUgll6v", "codice certificato" : "70621", "citta" : "sql1u_GhcznEZJ1", "nome" :
"8ksvZTBdlGtZAet", "sesso" : "femminile", "cognome" : "K_hSVNIwzd3c0BSZd2TqEjai2",
"telefono" : "tyPjCiEjU1", "data di Nascita" : "2018-04-02", "email" :
"GG7XIL8kwzT2KEjDAA2NwFwJHh3UnYUDd1jShXclkHwW8" }, "CentroSportivo" : { "idCentro" :
7, "nomeCentro" : "Lucen Sport" } }
Type "it" for more
```

QUERY DI CONFRONTO

-codiceFiscale,nome,cognome,dataNascita e nome dei CentroSportivo degli iscritti di sesso femminile

-VERSIONE MYSQL

```
mysql> select codiceFiscale,Iscritto.nome,cognome,dataNascita,CentroSportivo.nome from  
Iscritto,CentroSportivo where sesso="femminile" and Iscritto.idCentro=CentroSportivo.id;
```

codiceFiscale	nome	cognome	dataNascita	nome
00c003EMYGnCS2j	IFUrkyi9uOAxrGH	ypxtvh4ybFKa8pQjtync4DqCs	2020-03-21	Sport Center
00Nz6ga5Ognbd8p	pCW7OLpSut_s24P	D8FtzJGzojK6UstwRrqqn2pG4	2021-11-17	Sport Center
01SbcC99b5jloR1	nr7wamtFsDHOBa3	7FP7EdNw5kQFaOf25gM1Mw4hb	2020-11-03	Sport Center
01vYQyGG7E8yZr	rQ9HZZe531vnH7m	XR4woyQ7PBKsvlZcgpav0zXJF	2019-12-20	Sport Center

.....

_Y8P41gv1gPq9n1	TifDX9DSGFnFep4	9Q4hornPVw5OAE2OIxh5Y0n3j	2018-01-23	Appia Sport Village
_YCveteMv_liahl	FGNgyToQ7RiH6at	YXbPzEikhu1td57tVIWv7iE43	2018-10-07	Appia Sport Village
_yIXLHMVEIHvAA2	8fN1jICj92yevL2	laNKDHG3qY2IQyZOAuhnHOAjt	2021-07-09	Appia Sport Village
_z7jS957qTBQx4n	nWiwUARFsn651BE	jV4BvHm_MzUU5yYHu2EBslCg	2020-12-03	Appia Sport Village
_ZXTpLulsUoMchF	ypsoCDmFlmPAYM8	zMMylHJ9FmfPMQA5XGVvMhPiB	2021-08-10	Appia Sport Village
_h3Vp_VZaFZ0E0	b5QoHvYxcbe23e	Dq4TnsLEjWdAWNzAxfAbWiGI	2019-06-01	Appia Sport Village
_ryPijUHi6leb9	Qa2po5zU_Q4gH5D	7MOAf9i2jHoWN9WjySRNnTmVc	2021-02-11	Appia Sport Village

60093 rows in set (0.15 sec)

-conteggio degli iscritti di sesso femminile:

```
mysql> select count("codiceFiscale") from Iscritto,CentroSportivo where sesso="femminile";
```

count("codiceFiscale")
60093

1 row in set (0.12 sec)

-VERSIONE MONGODB

```
> db.Iscritto2.find({"dati iscritto.sesso":"femminile"}, {"dati iscritto.Codice Fiscale":1,"dati  
iscritto.nome":1,"dati iscritto.cognome":1,"CentroSportivo.nomeCentro":1});
```

```
{ "_id" : ObjectId("60e30d60fa545a7688f43955"), "dati iscritto" : { "nome" :  
"nx4tjLfaT7N795C", "cognome" : "8tCKoGTe5jhU7ZvdqwMs5nbpa" }, "CentroSportivo" :  
{ "nomeCentro" : "Sportessere" } }  
{ "_id" : ObjectId("60e30d60fa545a7688f43956"), "dati iscritto" : { "nome" :  
"q_WZNfve_r_mjDa", "cognome" : "Pw1wcd1QT3cRo1jF6tjRjv4Yv" }, "CentroSportivo" :  
{ "nomeCentro" : "Svago & Forma" } }  
{ "_id" : ObjectId("60e30d60fa545a7688f43957"), "dati iscritto" : { "nome" :  
"8ksvZTBdlGtZAet", "cognome" : "K_hSVNIwzd3c0BSZd2TqEjai2" }, "CentroSportivo" :  
{ "nomeCentro" : "Lucen Sport" } }  
{ "_id" : ObjectId("60e30d60fa545a7688f43958"), "dati iscritto" : { "nome" :  
"OJ87RH9qqRYD5FH", "cognome" : "4BPwmBOWngOZha90jTFJ0lfcP" }, "CentroSportivo" :  
{ "nomeCentro" : "Salerno Sport" } }  
{ "_id" : ObjectId("60e30d60fa545a7688f43959"), "dati iscritto" : { "nome" :  
"IFUrkyi9uOAxrGH", "cognome" : "ypxtvh4ybFKa8pQjtync4DqCs" }, "CentroSportivo" :  
{ "nomeCentro" : "Sport Center" } }
```

Type "it" for more

```
> var before=new Date();db.Iscritto2.find({"dati iscritto.sesso":"femminile"}, {"dati  
iscritto.Codice Fiscale":1,"dati iscritto.nome":1,"dati  
iscritto.cognome":1,"CentroSportivo.nomeCentro":1});var after=new Date();print((after-  
before)/1000+"s");
```

0.001s

-conteggio degli iscritti di sesso femminile:

```
> db.Iscritto2.count({"dati iscritto.sesso":"femminile"});
```

60093

Si osserva che il numero di iscritti di femminile è lo stesso sia per la tabella mysql che per la collection di mongoDB. Questo a testimoniare che lo script python, impiegando circa 15/20 minuti ha inserito valori dei record delle tabelle Iscritti e Centro Sportivo nella collection Iscritti2 in maniera corretta.

Dopodiché si può notare che il tempo di esecuzione della query in mongoDB è più veloce di quello di Mysql.

Infatti la query Mysql impiega 0.120 secondi contro i 0.001 secondi di mongoDB.

ALTRE QUERIES

-numero dei centri sportivi raggruppati per città e nome della città

```
> db.CentroSportivo.aggregate(
[{"$group":{"_id":"$indirizzo.citta","total":{"$sum":1}}},
{"$project":{"_id":0,"nomeCittà":"$_id","numeroCentri":"$total"}}]
);
```

```
{ "nomeCittà" : "Salerno", "numeroCentri" : 1 }
{ "nomeCittà" : "Roma", "numeroCentri" : 4 }
{ "nomeCittà" : "Frosinone", "numeroCentri" : 1 }
{ "nomeCittà" : "Velletri", "numeroCentri" : 1 }
{ "nomeCittà" : "Napoli", "numeroCentri" : 1 }
```

-numero degli iscritti per ogni centro sportivo con nome del centro

```
> db.Iscritto2.aggregate(
[{"$group":{"_id":"$CentroSportivo.nomeCentro","count":{"$sum":1}}},
{"$project":{"_id":0,"nomeCentro":"$_id","numeroscritti":"$count"}}]
);
```

```
{ "nomeCentro" : "Atletigame", "numeroscritti" : 12487 }
{ "nomeCentro" : "Country Club", "numeroscritti" : 12405 }
{ "nomeCentro" : "Sportessere", "numeroscritti" : 12387 }
{ "nomeCentro" : "Svago & Forma", "numeroscritti" : 12648 }
{ "nomeCentro" : "Sport Center", "numeroscritti" : 12557 }
{ "nomeCentro" : "Lucen Sport", "numeroscritti" : 12561 }
{ "nomeCentro" : "Appia Sport Village", "numeroscritti" : 12350 }
{ "nomeCentro" : "Salerno Sport", "numeroscritti" : 12622 }
```

Oss.

per fare quest'ultima query mongoDB ha impiegato 0.11s

```
> b=new Date();db.Iscritto2.aggregate([{"$group":
{"_id":"$CentroSportivo.nomeCentro","count":{"$sum":1}}},{"$project":
{"_id":0,"nomeCentro":"$_id","numeroscritti":"$count"}}]);print((new
Date()-b)/1000+"s");
0.11s
```

*mentre mysql ha impiegato 0.05 sec

```
mysql> select CentroSportivo.nome as nomeCentro,count("Iscritto.codiceFiscale") as  
numeroliscritti from CentroSportivo,Iscritto where Iscritto.idCentro=CentroSportivo.id  
group by CentroSportivo.id;
```

nomeCentro	numeroliscritti
Sport Center	12557
Sportessere	12387
Svago & Forma	12648
Atletigame	12487
Country Club	12405
Salerno Sport	12622
Lucen Sport	12561
Appia Sport Village	12350

8 rows in set (0.05 sec)

-media aritmetica di voto struttura,voto gestione e voto staff tecnico per ogni centro sportivo e nome di ogni centro sportivo

```
> db.CentroSportivo.aggregate([{"$project":  
{"nome":1,"_id":0,"votoS":"$recensioni.votoStruttura","votoG":"$recensioni.votoGestio  
ne","votoST":"$recensioni.votoStaffTecnico","votoFinale":{"$avg":  
["$recensioni.votoStruttura","$recensioni.votoGestione","$recensioni.votoStaffTecnico  
"]}}},{ "$project":{"votoS":0,"votoG":0,"votoST":0}},{ "$project":{"nome":1,"voto":  
{"$divide":[{"$trunc":{"$multiply":["$votoFinale",100]}}],100}}}]  
{ "nome" : "Sport Center", "voto" : 3.55 }  
{ "nome" : "Sportessere", "voto" : 4 }  
{ "nome" : "Svago & Forma", "voto" : 4 }  
{ "nome" : "Atletigame", "voto" : 3.33 }  
{ "nome" : "Country Club", "voto" : 3.33 }  
{ "nome" : "Salerno Sport", "voto" : 3.16 }  
{ "nome" : "Lucen Sport", "voto" : 2.83 }  
{ "nome" : "Appia Sport Village", "voto" : null }
```

*tempo impiegato:

```
> b=new Date();db.CentroSportivo.aggregate([{"$project":  
{"nome":1,"_id":0,"votoS":"$recensioni.votoStruttura","votoG":"$recensioni.votoGestio  
ne","votoST":"$recensioni.votoStaffTecnico","votoFinale":{"$avg":  
["$recensioni.votoStruttura","$recensioni.votoGestione","$recensioni.votoStaffTecnico  
"]}}},{ "$project":{"votoS":0,"votoG":0,"votoST":0}},{ "$project":{"nome":1,"voto":  
{"$divide":[{"$trunc":{"$multiply":["$votoFinale",100]}}],100}}}]];print((new Date()-  
b)/1000+"s")  
0.002s
```

-in seguito la versione mysql della query...

```
mysql> select Y.nome,truncate((votoStruttura+votoGestione+votoStaffTecnico)/3,2)
from (select X.idCentro,CentroSportivo.nome,truncate(tots/numR,2) as
votoStruttura,truncate(totG/numR,2) as votoGestione,truncate(totST/numR,2) as
votoStaffTecnico from (select count(idCentro) as numR,idCentro,sum(struttura) as
totS,sum(gestione) as totG,sum(staffTecnico) as totST from Recensione group by
idCentro) as X,CentroSportivo where idCentro=CentroSportivo.id group by idCentro) as
Y;
```

nome	truncate((votoStruttura+votoGestione+votoStaffTecnico)/3,2)
Sport Center	3.55
Sportessere	4.00
Svago & Forma	4.00
Atletigame	3.33
Country Club	3.33
Salerno Sport	3.16
Lucen Sport	2.83

7 rows in set (0.00 sec)