

Infinity Boulder

1. Abstract

"Infinity boulder" è una palestra di arrampicata situata nel Trentino. Come suggerisce il nome la palestra si specializza nel "Bouldering" una disciplina dell'arrampicata che si svolge su percorsi più corti ma più intensi. Lo sport si pratica senza essere legati a una corda ma piazzando sul suolo dei materassi di protezione. Infinity Boulder offre delle pareti artificiali alte 4,5 metri con appositi materassi. Su tali pareti sono costruiti dei percorsi con degli appigli o prese in plastica. La palestra offre circa 900 metri quadri di superficie arrampicabile divisa in 6 aree, in più ci sono anche zone dedicate allo yoga e all'allenamento.

Con il recente successo dello sport sono aumentate le persone che frequentano la palestra, inoltre per cause legate alla pandemia sono sorti nuovi bisogni ed esigenze. Per questi motivi la palestra ha deciso di aggiornare la sua base di dati. Questa base deve essere in grado di gestire tutti gli aspetti di una palestra come: iscrizioni, corsi, prenotazioni ma anche staff o l'inventario delle prese.

2. Analisi dei requisiti

2.1 Descrizione testuale

Nella base di dati sono presenti i dati degli **utenti** che sono coinvolti nella palestra di arrampicata, quali: nome, cognome, numero di telefono, luogo di nascita, data di nascita, e l'indirizzo.

Quest'ultimo è un attributo multi valore che comprende paese, città, provincia, CAP e indirizzo.

Un utente può essere iscritto alla palestra e frequentarla come cliente oppure far parte dello staff.

Per accedere alla palestra è obbligatorio essere iscritti alla Federazione Arrampicata Sportiva Italiana e possedere un certificato medico. Inoltre è possibile noleggiare le scarpette alla reception solo per un numero limitato di volte (6).

Di un **iscritto** sono noti: la scadenza del certificato medico, il numero di volte che ha noleggiato le scarpette, e il relativo codice della tessera FASI (Federazione Arrampicata Sportiva Italiana).

Un iscritto può avere quattro tipi di abbonamento: **annuale**, trimestrale o mensile oppure un pacchetto di **Ingressi**, in cui è memorizzato il numero di ingressi. La data è di interesse nell'abbonamento dei **3 mesi** oppure quello di **un mese**. Per l'abbonamento annuale non è necessaria la data in quanto scade a fine anno.

Ulteriori dettagli quali costo, nome, durata sono memorizzati nell'entità **dettagli pacchetto**.

Un iscritto per poter utilizzare la palestra deve prenotarsi tramite il sito della palestra, verranno memorizzate nella **prenotazione** le informazioni di durata, orario e data.

Un iscritto prenotato che si presenta in palestra occupa un'area della palestra, dunque viene inserito in **occupazione della palestra** di cui si ricorda anche l'orario di entrata, in tal modo si può sapere chi ha finito il proprio tempo di arrampicata. Le informazioni sull'occupazione della palestra sono costantemente aggiornate dallo staff.

Un iscritto può essere un **agonista** i cui dati sono: categoria, scheda, altezza, ape index.

L'agonista può partecipare a una **competizione** i cui dati significativi sono: nome di campionato, classifica, numero zone, numero top, luogo e data della competizione.

Una competizione si svolge su 5 **route** (percorsi) di arrampicata che si caratterizzano per: inclinazione del muro, tipo di presa, dinamicità del percorso, il numero della route che identifica il percorso. Questi dati sono tenuti al fine di analizzare i punti di forza o debolezza degli atleti ed allenare al meglio gli agonisti.

Un iscritto può svolgere un **corso** le cui informazioni sono: il codice del corso, l'età per cui il corso è destinato, la difficoltà, ora e il giorno.

Un corso è tenuto da un trainer, che fa parte dello staff, così come il routesetter e il receptionist; di quest'ultimo si conoscono i turni.

Dello **staff** si conosce lo stipendio.

Il **routesetter**, di cui si conosce il soprannome, gestisce un'area della palestra tra quelle dedicate all'arrampicata.

L'**area**, le cui informazioni sono nome e capienza, indica una delle aree dedicate all'arrampicata, ai pesi o ad altro ancora.

L'area **pesi** contiene l'ID del peso in questione, la marca, e il peso effettivo in chilogrammi.

Le aree di arrampicata contengono delle **route** realizzate dai routesetter, ogni route ha un l'ID percorso, una data della tracciatura, la difficoltà e il colore delle prese.

I percorsi sono composti dalle **prese in uso**.

Le **prese in magazzino** possono dover essere aggiustate o lavate prima di essere di nuovo pronte per l'utilizzo.

Le prese sono memorizzate nell'**inventario prese** che è costituito dalle prese in uso e da quelle in magazzino le cui informazioni sono ID presa, colore, marca, materiale.

2.2 Glossario dei termini

Termine	Descrizione	Collegamenti
Utente	Qualsiasi persona che è stata registrata nella palestra	Iscritto, Staff
Iscritto	Utenti iscritti, tutti gli utenti che vogliono accedere alla palestra devono essere iscritti, l'iscrizione deve essere rinnovata ogni anno.	Agonista, Prenotazione, Occupazione palestra, Utente
Prenotazione	Indica lo slot di prenotazione,	Iscritto, Occupazione palestra
Occupazione palestra	Lista di iscritti con rispettiva prenotazione che al momento occupano la palestra	Iscritto, Prenotazione, Area
Area	Insieme delle aree della palestra, ne fanno parte le aree dove si arrampica ma anche le sale dedicate a pesi o allo yoga	Occupazione palestra, Peso, Route, Corso, Routesetter
Peso	Insieme di tutti i pesi posseduti dalla palestra	Area
Route	Percorso da arrampicare che si trova in una delle aree della palestra. E' realizzato tramite delle prese in plastica fissate alla parete. Si dice che un percorso è "tracciato" dal routesetter.	Area, Routesetter, Presa in uso
Prese in uso	Insieme delle prese (ovvero appigli) che in quel momento sono usati nei percorsi della palestra	Route
Prese in magazzino	Insieme delle prese che al momento non sono usate	
Inventario prese	Insieme di tutte le prese possedute dalla palestra	
Agonista	Atleta agonista, partecipa a competizioni e rappresenta la palestra	Iscritto, Competizione
Competizione	Gara di arrampicata, si basa su chi completa il maggior numero di route	Agonista, Route di gara
Route di gara	Insieme di route delle varie gare fatte dagli atleti	Competizione
Corso	Insieme dei corsi che la palestra offre, si svolgono in un area della palestra	Iscritto, Area

Termine	Descrizione	Collegamenti
Staff	Insieme delle persone che lavorano nella palestra	Utente
Routesetter	Persona che realizza Route (percorsi) della palestra fissando le prese al muro	Area, Route
Trainer	Allenatore, insegna corsi	Corso
Receptionsita	Persona alla cassa che gestisce iscrizioni, ingressi, abbonamenti.	
Annuale	Iscritto con abbonamento annuale, scade a fine anno	Dettagli pacchetto
3 mesi	Iscritto con abbonamento da 3 mesi, scade dopo 90 giorni dal giorno del pagamento	Dettagli pacchetto
1 mese	Iscritto con abbonamento da 2 mese, scade dopo 30 giorni	Dettagli pacchetto
10 ingressi	Iscritto che ha comprato un pacchetto da 10 ingressi, devono essere utilizzati entro la fine dell'anno corrente	Dettagli Pacchetto
Dettagli Pacchetto	Contiene le informazioni sui pacchetti che un iscritto può comprare ovvero sugli abbonamenti e ingressi.	Annuale, 1 mese, 3 mesi, ingressi

2.3 Operazioni tipiche

Descrizione dell'operazione	Numero ripetizioni
Un utente fa l'iscrizione	200 al mese
E' comprato un abbonamento mensile	150 al mese
Un iscritto con prenotazione accede alla palestra	300 al giorno
Ricerca del routesetter per un determinato percorso	100 al mese
Ricerca delle Route di gara effettuate da un agonista, ordinate per angolo della parete	150 all'anno
Ricerca dei corsi insegnati da un trainer	350 al mese
Calcolare la somma delle persone in una determinata area della palestra	300 al giorno
Un iscritto esce dalla palestra	300 al giorno
Calcolare il numero di persone iscritte a ciascun corso	100 all'anno

Descrizione dell'operazione	Numero ripetizioni
Trovare il numero di percorsi di una certa difficoltà	100 all'anno
Viene scalato un ingresso da un iscritto che possiede il pacchetto da 10 ingressi	300 al mese
Un iscritto effettua una prenotazione	100 al giorno

3 Progettazione concettuale

3.1 Lista entità e descrizione attributi

Utente		
<u>CE</u>	varchar(16) primary key	Codice fiscale della persona
Nome	varchar(30)	
Cognome	varchar(30)	
Telefono	int	
Data di nascita	date	
Luogo di nascita	varchar(30)	
Paese	varchar(30)	Paese di residenza
Città	varchar(30)	Città di residenza
Provincia	varchar(2)	Provincia di residenza
CAP	varchar(5)	
Indirizzo	varchar(50)	Via di residenza
Iscritto		
ScadenzaCertmedico	date	Indica la data di scadenza del certificato medico
NumNoleggioScarpette	int	Indica il numero di volte che l'utente ha noleggiato le scarpette, non si possono noleggiare più di 3 volte
<u>CodTessera</u>	varchar(10) primary key	Numero identificativo riportato nella tessera F.A.S.I. ,è obbligatorio avere la tessera per accedere alla palestra, la tessera scade a fine anno

Iscritto si specializza in annuale, 3 mesi, 1 mese, 10 ingressi:

annuale		
3 mesi		
data	date	data in cui scade l'abbonamento

1 mese		
data	date	data in cui scade l'abbonamento
Ingressi		
NumIngressi	int	numero degli ingressi in palestra effettuati
DettagliPacchetto		
costo	float	il costo del pacchetto scelto
durata	varchar(100)	note sulla durata precisa del pacchetto
<u>nome</u>	varchar(30) primary key	nome del pacchetto scelto (annuale, trimestrale, mensile, 10 ingressi)
OccupazionePalestra		
OrarioEntrata	time	orario in cui la persona entrata in palestra
Prenotazione		
Durata	time	Tempo per cui la persona può stare in palestra
<u>Orario</u>	time primary key	Orario in cui la persona dovrebbe arrivare in palestra per il suo turno
<u>Data</u>	date primary key	Giorno della prenotazione
Area		
<u>Nome</u>	varchar(30) primary key	nome dell'area della palestra
Capienza	int	numero di persone massimo che quell'area può contenere
Peso		
<u>IDPeso</u>	varchar(5) primary key	identificativo del peso
Marca	varchar(15)	Marca produttrice del peso
Pesokg	float	Peso effettivo in chilogrammi

Route		
DataTracciatura	date	Data in cui il percorso è stato realizzato, serve a tenere traccia dei percorsi più vecchi. Questi saranno i primi a essere ritracciati o cambiati.
Difficoltà	varchar(1)	Lettere dell 'alfabeto da A ad F che indicano la difficoltà del percorso. La lettera A indica i percorsi più facili la F i percorsi più difficili.
Colore	varchar(30)	Colore del percorso
<u>ID route</u>	varchar(10) primary key	Identificativo del percorso

Inventario prese		
<u>ID presa</u>	varchar(10) primary key	Identificativo della presa
Colore	varchar(30)	Colore della presa
Marca	varchar(30)	Marca produttrice della presa
Materiale	varchar(30)	Materiale di cui è fatta la presa, può variare da plastica a legno o anche fibra di vetro

Inventario prese si specializza in presa in uso e presa in magazzino:

Presa in uso		
Presa in magazzino		
DaLavare	bool	Indica se la presa è stata lavata o no. Prima di essere riusate le prese vengono lavate.
Rotta	bool	Indica se la presa è rotta, se il danno è lieve a volte è rivenduta o utilizzata

Staff		
stipendio	float	stipendio del lavoratore

Staff si specializza in routesetter, trainer, receptionista.

Routesetter		
soprannome	varchar(30)	Nome del routesetter (tracciatore)
Trainer		
Receptionista		
Turni	varchar(50)	Indica gli orari e i giorni nei quali l'utente lavora alla reception, turno 1 è dalle 12:30 alle 17:30, turno 2 è dalle 17:30 alle 22:30

Corso		
Giorni	varchar(30)	giorni della settimana in cui si svolge il corso, per esempio Mar-Gio indica che il corso si svolge il martedì e il giovedì
Orario	varchar(30) primary key	fascia oraria in cui si svolge il corso
Difficoltà	varchar(30)	indica a chi è rivolto il corso, per esempio può essere un corso base per principianti o avanzato per esperti
Età	varchar(30)	Età a cui è dedicato il corso, è riportata sia l'età minima che quella massima (es 14-16 anni)
CodCorso	varchar(10) primary key	identificativo del corso

Agonista		
Categoria	varchar(10)	Categoria in cui gareggia l'agonista
Scheda	varchar(30)	Nome della scheda di allenamento che l'atleta segue. E' riportato solo il nome, la scheda effettiva è fornita dal trainer
Altezza	int	altezza in centimetri dell'atleta
ApelIndex	float	misura della lunghezza delle braccia in relazione all'altezza, misurata con le braccia aperte. (es +4 indica che l'apertura delle braccia è 4cm in più rispetto all'altezza)
Competizione		
nome campionato	varchar(30)	nome del campionato a cui appartiene la gara (es campionato regionale o italiano)
Classifica	int	Posizione in cui l'atleta si è classificato a fine gara
NumTop	int	La competizione si svolge su 5 percorsi con l'obiettivo di arrivare in cima a tutti. Il numero di Top indica quanti percorsi sono stati completati
NumZone	int	All'atleta sono attribuiti punti anche quando arriva a metà del percorso anche se poi non lo termina, il numero di zone indica in quanti percorsi l'atleta è arrivato a metà

Competizione		
<u>Luogo</u>	Varchar(30) primary key	Nome della palestra dove si è svolta la competizione
<u>Data</u>	date primary key	giorno in cui si è svolta la competizione
Route di gara		
<u>NumeroRoute</u>	int [1,5] primary key	La competizione si svolge su 5 percorsi o route. Sono solitamente numerati da 1 a 5 per identificarli.
Dinamicità	varchar(30)	Indica quanto dinamico è un percorso, si fa riferimento allo stile di arrampicata del percorso
TipoDiPresa	varchar(30)	Indica lo stile di prese usate nel percorso, anche questo attributo identifica un certo stile di arrampicata
InclinazioneMuro	int	indica l'inclinazione della parete su cui è stato tracciato il percorso, diverse inclinazioni comportano stili di arrampicata molto diversi
Completamento	float	indica se l'atleta ha completato o no il percorso: - 1 = top, percorso completato; - 0,5 = zone, percorso completato a metà; - 0 = percorso non completato nemmeno a metà;

3.2 Descrizione delle relazioni

Relazione	Entità coinvolte	Descrizione
UtenteIscritto	Utente (0,1) Iscritto (1,1)	Un utente diventa un iscritto nel momento in cui ha una tessera, un iscritto è per forza un utente. Per accedere alla palestra bisogna essere iscritti.
UtenteStaff	Utente (0,1) Staff (1,1)	Un utente della palestra può far parte dello staff, un elemento dello staff è di conseguenza un utente
Effettuazione	Iscritto (0,N) Prenotazione (0,6)	Un iscritto può effettuare o meno una prenotazione, una prenotazione può essere stata effettuata da al massimo 6 iscritti diversi
Prevedere	Prenotazione (0,6) Occupazione Palestra (1,1)	La prenotazione prevede l'occupazione della palestra, se la palestra è occupata c'è una prenotazione. E' necessario avere una prenotazione per l'orario giusto per entrare nella palestra.
IscrittoAgonista	Iscritto (0,1) Agonista (1,1)	Un iscritto può essere un agonista, un agonista invece è per forza iscritto.

Relazione	Entità coinvolte	Descrizione
Partecipazione	Agonista (0,N) Competizione (1,1)	Un agonista può partecipare a più competizioni, in una competizione partecipa solo un atleta
RouteCompetizione	Route di gara (1,1) Competizione (1,5)	Una route di gara è sicuramente parte di una competizione di arrampicata, una competizione si svolge su da una a 5 route di gara
Insegnamento	Corso (1,1) Trainer (1,N)	Un corso è tenuto da un trainer, mentre un trainer insegna almeno un corso
Realizzazione	Routesetter (1,N) Route (1,1)	Il route setter può realizzare più route, una route è realizzata da un solo routesetter
Composizione	Route (1, N) Presa in uso (1,1)	Una route può essere composta da più prese, una presa fa parte di una precisa route
Gestione	Routesetter (1,N) Area (1,N)	Il route setter gestisce una o più aree, un'area è gestita da uno o più routesetter
Svolgimento	Corso (1,1) Area (0,N)	Un corso si svolge in una precisa area, mentre in un'area si possono svolgere diversi corsi
IscrittoOccupazione Palestra	Iscritto (0,1) Occupazione Palestra (1,1)	Un iscritto può occupare la palestra, e l'occupazione della palestra memorizza l'iscritto interessato.
OccupazionepalestraArea	Occupazione palestra (1,1) Area (0,N)	L'occupazione della palestra riguarda un'area, un'area può essere occupata da diverse persone/occupazioni
Contenere	Area (0,N) Peso (1,1)	L'area può contenere diversi pesi, mentre un peso è contenuto nella sua area
Contenere	Route (1,1) Area (0,N)	L'area può contenere diverse route, le route sono contenute solo in un area
DettagliPacchetto Annuale	DettagliPacchetto (1,1) Annuale (1,1)	L'abbonamento annuale è personale e rimanda ai dettagli di tale pacchetto. Un abbonamento annuale ha dei dettagli, mentre i dettagli riguardano un abbonamento.
DettagliPacchetto 3Mesi	DettagliPacchetto (1,1) 3Mesi (1,1)	Un abbonamento trimestrale ha delle specifiche, i dettagli riguardano un abbonamento.
DettagliPacchetto 1Mese	DettagliPacchetto (1,1) 1Mese (1,1)	Un abbonamento mensile rimanda a dei dettagli, i dettagli riguardano un abbonamento.
DettagliPacchetto 10Ingressi	DettagliPacchetto (1,1) 10Ingressi (1,1)	Un pacchetto di 10 ingressi ha dei dettagli specifici, i dettagli riguardano il pacchetto in questione.

Vincoli non rappresentabile tramite lo schema ER

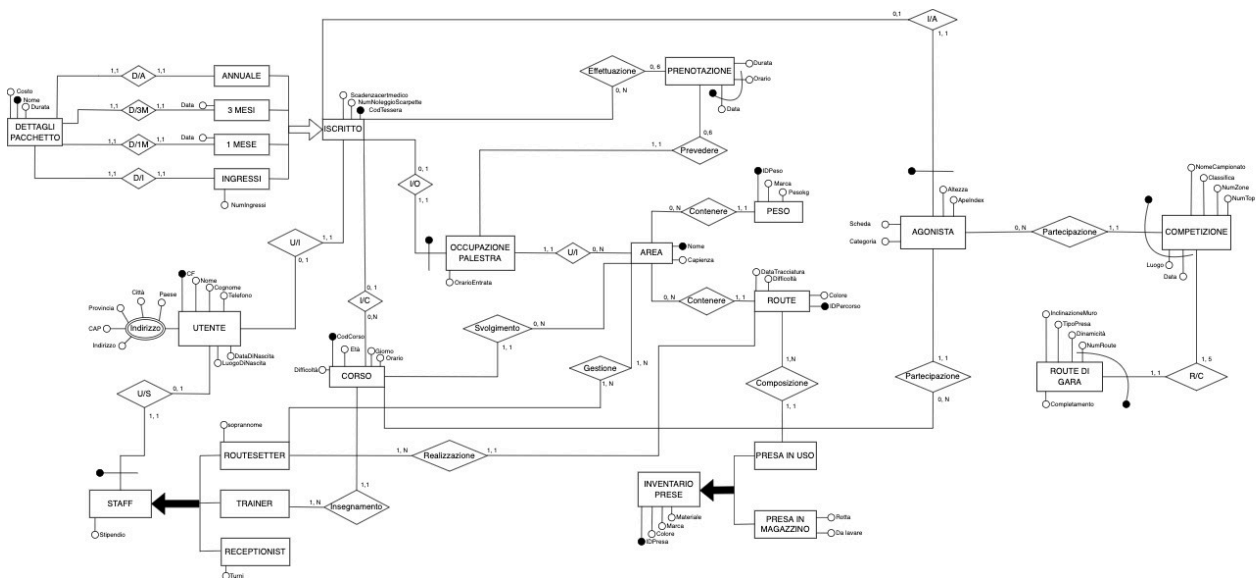
- Un iscritto può occupare la palestra solo nel suo slot di prenotazione (dall'orario di prenotazione per un periodo pari alla durata della prenotazione)

Vincoli di derivazione

- L'attributo NumTop può essere calcolato sommando le route di gara dove il completamento è uguale a 1;

- L'attributo NumZone può essere calcolato sommando le route di gara dove il completamento è 0.5;

3.3 Schema concettuale



4 Progettazione Logica

4.1 Ristrutturazione

4.1.1 Analisi delle ridondanze

L'attributo NumTop in competizione può essere dedotto dall'attributo completamento in route di gara. Infatti il numero di top corrisponde al numero di route completate, più precisamente corrisponde a contare il numero di route di gara dove l'attributo completamento è 1.

Concetto	Tipo	Volume
Competizione	E	1000
Route di gara	E	5000
Svolgimento (R/C)	R	5000

Consideriamo due operazioni tipiche:

1. Viene aggiunta una route di gara che è stata completata dall'atleta. 200 volte al mese.
2. Si vuole leggere il NumTop di un atleta in una determinata competizione. 500 volte al mese.

Presenza di ridondanza

Operazione 1: bisogna inserire la route completata nell'entità route di gara, successivamente va aggiornato il valore di NumTop in Competizione.

Concetto	Costrutto	Accessi	Tipo
Route di Gara	E	1 (x200)	S
Svolgimento	R	1 (x200)	L
Competizione	E	1 (x200)	S

Operazione 2: basta leggere l'attributo NumTop in Competizione.

Concetto	Costrutto	Accessi	Tipo
Competizione	E	1 (x500)	L

assumiamo che le operazioni in scrittura costino il doppio di quelle in lettura, il costo mensile totale allora sarà:

$$200 \times 2 + 200 + 200 \times 2 + 500 = 1500$$

Assenza di ridondanza: assumiamo che l'attributo "NumTop" non ci sia

Operazione 1: Basta inserire la route completata nell'entità route di gara

Concetto	Costrutto	Accessi	Tipo
Route di Gara	E	1 (x200)	S

Operazione 2: Bisogna andare a trovare tutte le route (5 per competizione) di una determinata competizione e contarle.

Concetto	Costrutto	Accessi	Tipo
Competizione	E	1 (x500)	L
Svolgimento	R	5 (x500)	L
Route di gara	E	5 (x500)	L

assumiamo che le operazioni in scrittura costino il doppio di quelle in lettura, il costo mensile totale allora sarà:

$$200 \times 2 + 500 + 500 \times 5 + 500 \times 5 = 5900$$

Possiamo quindi concludere che è conveniente mantenere la ridondanza. Lo stesso ragionamento analogo può essere fatto per NumZone per cui la situazione è identica, dunque manterremo anche questa ridondanza.

4.1.2 Eliminazione delle generalizzazioni

Generalizzazione	Risoluzione
Iscritto ← Annuale, 3 mesi, 1 mese, Ingressi	<ul style="list-style-type: none"> • Si accorpano le entità <i>Annuale</i>, <i>3 mesi</i>, <i>1 mese</i> in una nuova entità chiamata <i>Abbonamento</i> e vengono aggiunti gli attributi <i>tipo</i> varchar(30) primari key, <i>data</i> date. • Si mantiene l'entità <i>Ingressi</i> • Si aggiunge una terza entità <i>DettagliPacchetto</i> che avrà gli attributi <i>costo</i> float, <i>nome</i> varchar(30) primari key e <i>Durata</i> varchar(30)
Staff ← Routesetter, Trainer, Receptionist	<p>Per risolvere generalizzazione si è deciso di accorpare il padre nei figli.</p> <p>Si elimina l'entità <i>Staff</i> e si sostituisce la generalizzazione con delle relazioni da utente (0,1) a <i>Routesetter</i> (1,1), <i>Trainer</i> (1,1), <i>Receptionist</i> (1,1). L'attributo <i>stipendio</i> viene assegnato a ogni entità coinvolta.</p> <p>La chiave è esterna verso Utente.</p>
Inventario prese ← Presa in uso, Presa in magazzino	<p>Per ristrutturare la generalizzazione si è scelto di accorpare i figli nel padre.</p> <p>Le entità <i>Presa in uso</i> e <i>Presa in magazzino</i> vengono accorpate al padre <i>Inventario prese</i> e vengono aggiunti come attributi <i>in uso</i> bool, <i>in magazzino</i> bool, <i>rotta</i> bool, <i>da lavare</i> bool.</p> <p>La chiave rimane ID presa.</p>

Anche l'attributo composto *Indirizzo* è stato modificato, assegnando gli attributi *paese*, *provincia*, *città*, *CAP*, *Indirizzo* (*via* e *numero civico*) all'entità *Utente*.

4.1.3 Scelta degli identificatori primari

Sia *Abbonamento* che *Ingressi* presentano una chiave esterna verso *Iscritto*.

Utente → Utente.CF
Receptionist (Utente, Stipendio, Turni)
 Utente → Utente.CF
Corso (CodCorso, Area, Trainer, Giorno, Orario, Difficoltà, Età)
 Area → Area.Nome
 Trainer → Trainer.Utente
Prenotazione (Orario, Data, Durata)
Effettuazione (Data, Orario, Iscritto)
 Data → Prenotazione.Data
 Orario → Prenotazione.Orario
 Iscritto → Iscritto.CodTessera
Occupazione Palestra (Iscritto, OraPrenotazione, DataPrenotazione, OrarioEntrata)
 Iscritto → Iscritto.CodTessera
 DataPrenotazione → Prenotazione.Data
 OraPrenotazione → Prenotazione.Orario
Area (Nome, Capienza)
Peso (IDPeso, Area, Marca, Pesokg)
 Area → Area.Nome
Route (IDPercorso, Routesetter, Area, Colore, Difficoltà, DataTracciatura)
 Routesetter → Routesetter.Soprannome
 Area → Area.Nome
Inventario prese (IDPresa, Route, Colore, Marca, Materiale, InUso, InMagazzino, Rotta, DaLavare)
 Route → Route.IDPercorso

5 Query e indici

5.1 Query

1) Dato il codice fiscale dell'agonista LZZGDU94D07B888Z, mostrare il completamento delle route di gara che ha tentato ordinate secondo l'inclinazione del muro.

```

Select InclinazioneMuro, Completamento
from RouteDiGara
where Agonista in ( Select CodTessera
                    From Utente as u join Iscritto as i on i.CF=u.CF
                    where u.CF='LZZGDU94D07B888Z')
Order by InclinazioneMuro;
  
```

	inclinazionemuro integer	completamento double precision
1	0	0.5
2	20	0
3	30	1
4	35	1
5	50	1

2) Per ogni altezza degli agonisti trovare le somme delle vie completate, raggruppate per tipo di presa ordinato per altezza

```

select Altezza, TipoPres, sum(Completamento)
from Agonista as a join RouteDiGara as r on a.CodTessera=r.Agonista
group by Altezza, TipoPres
order by Altezza
  
```

	altezza integer	tipopresa character varying (30)	sum double precision
1	153	Misto	1
2	153	Svasi	0.5
3	153	tacche	2
4	164	Buchi	0.5
5	164	Misto	0.5
6	164	Svasi	0.5
7	164	Tacche	0.5
8	172	Buchi	1
9	172	Misto	0.5
10	172	Svasi	1
11	172	Tacche	0.5
12	175	Misto	1
13	175	Svasi	0
14	175	Tacche	0.5
15	178	Buchi	0.5
16	178	Misto	1
17	178	Svasi	1
18	178	Tacche	1

3) Trovare il materiale di presa più usato da ogni routesetter e il loro corrispettivo stipendio

```

Select Routesetter, MaterialePiùUsato, Stipendio
from Routesetter join (select Routesetter,max(Materiale) as MaterialePiùUsato
from InventarioPrese,Route
where InventarioPrese.Route = Route.IDPercorso
group by Routesetter) as RM on Routesetter.Soprannome=RM.Routesetter

```

	routesetter character varying (30)	materialepiùusato text	stipendio double precision
1	Tony	Fibra di vetro	1490
2	Jonny	Plastica	1030
3	Ken	legno	1380
4	Penny	Plastica	1050

4) Trovare gli orari di prenotazione in cui la media del numero del noleggio delle scarpette è maggiore di 3

```

select Orario
from Iscritto as i join Effettuazione as e on i.CodTessera=e.Iscritto
group by e.Orario
having avg(NumNoleggioScarpette) > 3

```

	orario time without time zone
1	18:30:00
2	19:30:00
3	19:00:00
4	20:00:00
5	18:00:00

5) Trovare gli iscritti che partecipano al corso "INT01" e che hanno un abbonamento mensile.

```

select Abbonamento.CodTessera
from Abbonamento join Iscritto on Abbonamento.CodTessera = Iscritto.CodTessera
where Iscritto.CodCorso='INT01' and Abbonamento.Nome='mensile'

```

	codtessera [PK] character varying (10)
1	12469

6) Trovare nome e cognome delle persone che frequentano i corsi di trainer con uno stipendio maggiore di 1730

```

select Nome,Cognome
from Iscritto join Utente on Iscritto.CF=Utente.Cf
where Iscritto.CodCorso in (select CodCorso
from Corso join Trainer on Corso.Trainer=Trainer.Utente
where Trainer.Stipendio > 1730)

```

	nome character varying (30)	cognome character varying (30)
1	Ippolito	Calvello
2	Gherardo	Rosolino
3	Libero	Legato
4	Ivan	Tufanisco
5	Angelo	Passera
6	Amone	Maggiani
7	Isidoro	Varischi
8	Guido	Lazzaro
9	Guglielmo	Ruzzanti
10	Anna	Cifone
11	Gustavo	Tasco

5.2 Indici

Come si può vedere dalle operazioni tipiche si accede spesso in lettura alla relazione Route, invece l'accesso in scrittura è fatto raramente, inoltre in una palestra di grandi dimensioni il numero

di route sarà abbastanza grande. Per queste ragioni è opportuno considerare l'uso di un indice sulla relazione Route.

```
create index idx_Route on Route(IDPercorso);
```

6 Codice C++

6.1 Descrizione dell'utilizzo e spiegazione del codice

Il programma c++ creato riproduce le query sopra elencate, con la possibilità aggiunta di interagire e cambiare alcuni parametri di input sulle query come il codice fiscale o lo stipendio.

Per eseguire il programma è necessario avere nella stessa directory il sorgente c++ e una cartella dependencies con dentro tutte i file necessari della libreria che ci permetterà di accedere al database via software (Per indicazioni precise si faccia riferimento all'apposito laboratorio del corso). Infine bisogna creare l'eseguibile da terminale, navigando nella directory del file sorgente e inserendo il comando:

```
c++ main.cpp -L dependencies/lib -lpq -o main
```

Per il funzionamento corretto del programma bisogna modificare il file c++ inserendo manualmente le informazioni sul database locale nell'apposita sezione.

```
#define PG_HOST "127.0.0.1"
#define PG_USER "filippotonini" // nome utente
#define PG_DB "Progetto" //il nome del database
#define PG_PASS "xxxxxx" // password
#define PG_PORT 5432
```

Il programma stampa una descrizione a parole delle query e chiede all'utente di selezionare un numero tra 1 e 6, ognuno corrisponde a una query diversa. Digitando 0 il programma termina. Se la query scelta lo richiede sono chiesti ulteriori input all'utente.

Una volta scelti tutti gli input il programma stampa in forma di tabella il risultato della query e torna alla selezione delle query.

Il programma effettivo contiene:

- la funzione **connectDB** al database tramite la funzione **PQconnectdb(conninfo)** che permette di accedere al database usando le informazioni sul database;
- una funzione **checkResults** che permette di controllare che la query restituisca le tuple in modo corretto tramite la funzione **PQresultStatus(res);**
- una funzione **PrintQuery** che stampa le query su standard output. Per far sì che la stampa sia allineata è necessario calcolare la lunghezza massima di ogni colonna. Per fare ciò è necessario salvare il risultato della query in una matrice che rappresenterà la tabella da stampare, contenente sia gli attributi che le tuple, successivamente si scorre ogni colonna della matrice per trovare la cella con lunghezza massima. Infine si può procedere alla stampa effettiva, alle celle con lunghezza minore della massima sono aggiunti degli spazi vuoti. Per rendere la stampa più ordinata è stata usata una funzione ausiliaria **PrintLine;**
- il main che contiene: una prima stampa della descrizione delle query; la scelta delle query tramite input di un numero tra 1 e 6; uno switch che, a seconda dei diversi casi, comunica la query scelta al database tramite la funzione **PQexec** (per alcune query sono chiesti ulteriori input dall'utente); la chiusura della connessione col database tramite **PQfinish.**