

UNIVERSITÀ DELLA CALABRIA
DIP. DI MATEMATICA E INFORMATICA

CORSO DI LAUREA IN INFORMATICA
A.A. 2020/2021

Progetto per il Corso di
BASI DI DATI

Gruppo <8>

<210388, Torchia, Salvatore>

<209508, Fiorentino, Salvatore>

<213622, Marasco, Giovanni>

<209570, Cuteri, Andrea>

<209555, Lorenti, Domenico>

DOCENTE

Prof. P. Rullo

ESERCITATORI

Ing. G. Labocchetta

Ing. Marco Mastratise

1 Tematica Progettuale

La progettazione del sistema informatico in esame riguarda la base di dati per la gestione dei dati relativi ad un centro fitness.

2 Raccolta e Analisi dei Requisiti

2.1 Raccolta dei requisiti

	Requisiti Richiesti
	<p>Per il centro di fitness lavora uno staff, costituito da personale amministrativo, medici e da un gruppo di istruttori. Ogni membro dello staff è caratterizzato da nome, cognome, sesso, luogo e data di nascita.</p> <p>Un medico è caratterizzato inoltre dalla data di conseguimento della abilitazione alla professione e dalla data di conseguimento del titolo.</p> <p>Un istruttore può avere il titolo di diploma ISEF o di laurea in Scienze Motorie può avere conseguito uno o più brevetti. Un brevetto è identificato da un titolo, da un livello, una descrizione, è emesso da un ente sportivo e dà una qualifica per l'insegnamento di un determinato tipo di attività sportiva (nuoto, tennis, body-building...). Si vuole conoscere tutti i brevetti posseduti da un istruttore e la data di conseguimento degli stessi.</p> <p>Il database deve tenere traccia dei clienti (o soci) del centro di fitness. Un cliente è caratterizzato da nome, cognome, sesso, luogo e data di nascita.</p> <p>Un abbonamento è caratterizzato da una data di inizio e da una durata di validità ed fornisce l'iscrizione del cliente a una o più attività sportive. Le attività sportive possono essere svolte in sala o in un campo. Il costo dell'abbonamento dipende dal costo delle singole attività sportive scelte dal cliente.</p> <p>Alla scadenza del proprio abbonamento, il cliente può rinnovare lo stesso o sottoscriverne uno nuovo (ad esempio, con attività sportive diverse). Il cliente deve sottoporsi a una visita medica, effettuata da uno dei medici del centro, sia al momento dell'iscrizione che del rinnovo dell'abbonamento.</p> <p>Un'attività sportiva è caratterizzata da un nome univoco e da una descrizione. Si vuole memorizzare, inoltre, il numero massimo di partecipanti e il numero di iscritti.</p> <p>Il centro di fitness si compone di un certo numero di sale, di diversa tipologia (sala fitness, sala corsi), da uno o più spogliatoi maschili e femminili e da un certo numero di campi da gioco.</p>

Per sale e per campetti, si vuole memorizzare il nome, univoco all'interno del fitness center, una descrizione e la capienza (espressa in numero di persone).

Per tutti i tipi di attività sportiva sono previste una o più lezioni settimanali, ognuna delle quali è caratterizzata da un giorno, un orario di inizio e una durata, espressa in minuti.

Per ogni lezione, si vuole memorizzare la sala o il campetto in cui si svolge e l'istruttore che la supervisiona.

Un cliente, inoltre, può prenotare un campetto (negli orari in cui questo non è occupato per le lezioni). A tal fine, è di interesse conoscere anche il costo per la prenotazione di ogni campetto da gioco.

2.2 Analisi dei requisiti

2.2.1 Eliminazione delle Ambiguità

LINEA	TERMINE	SINONIMI	MOTIVAZIONE CORREZIONE
1	Cliente (per la prenotazione del campo)	Persona	Quando ci si riferisce ad un cliente nel contesto della prenotazione di un campo quest'ultimo non deve necessariamente aver sottoscritto un abbonamento.
2	campo da gioco	campetto, campetto da gioco	i campetti sono i campi da gioco utilizzabili per le lezioni oppure prenotabili quando non sono occupati dalle lezioni
3	sottoscrizione (quando ci si riferisce all'abbonamento)	iscrizione	un abbonamento viene sottoscritto per la prima volta e poi viene rinnovato (non è lo stesso concetto di iscrizione all'attività sportiva)
4	laurea	titolo del medico	un medico deve sempre aver conseguito la laurea

2.2.2 Ristrutturazione dei Requisiti Richiesti

	SPECIFICHE RISTRUTTURATE
	<p>Per il centro di fitness lavora uno staff, costituito da personale amministrativo, medici e da un gruppo di istruttori. Ogni membro dello staff è caratterizzato da nome, cognome, sesso, luogo e data di nascita.</p> <p>Un medico è caratterizzato inoltre dalla data di conseguimento della abilitazione alla professione e dalla data di conseguimento della laurea.</p> <p>Un istruttore può avere il titolo di diploma ISEF o di laurea in Scienze Motorie può avere conseguito uno o più brevetti. Un brevetto è identificato da un titolo, da un livello, una descrizione, è emesso da un ente sportivo e dà una qualifica per l'insegnamento di un determinato tipo di attività sportiva (nuoto, tennis, body-</p>

	<p>building...). Si vuole conoscere tutti i brevetti posseduti da un istruttore e la data di conseguimento degli stessi.</p> <p>Il database deve tenere traccia dei clienti (o soci) del centro di fitness. Un cliente è caratterizzato da nome, cognome, sesso, luogo e data di nascita.</p> <p>Un abbonamento è caratterizzato da una data di inizio e da una durata di validità ed fornisce l'iscrizione del cliente a una o più attività sportive. Le attività sportive possono essere svolte in sala o in un campo da gioco. Il costo dell'abbonamento dipende dal costo delle singole attività sportive scelte dal cliente.</p> <p>Alla scadenza del proprio abbonamento, il cliente può rinnovare lo stesso o sottoscriverne uno nuovo (ad esempio, con attività sportive diverse). Il cliente deve sottoporsi a una visita medica, effettuata da uno dei medici del centro, sia al momento della sottoscrizione che del rinnovo dell'abbonamento.</p> <p>Un'attività sportiva è caratterizzata da un nome univoco e da una descrizione. Si vuole memorizzare, inoltre, il numero massimo di partecipanti e il numero di iscritti. Il centro di fitness si compone di un certo numero di sale, di diversa tipologia (sala fitness, sala corsi), da uno o più spogliatoi maschili e femminili e da un certo numero di campi da gioco.</p> <p>Per sale e per i campi da gioco, si vuole memorizzare il nome, univoco all'interno del fitness center, una descrizione e la capienza (espressa in numero di persone).</p> <p>Per tutti i tipi di attività sportiva sono previste una o più lezioni settimanali, ognuna delle quali è caratterizzata da un giorno, un orario di inizio e una durata, espressa in minuti.</p> <p>Per ogni lezione, si vuole memorizzare la sala o il campo da gioco in cui si svolge e l'istruttore che la supervisiona.</p> <p>Una persona, inoltre, può prenotare un campo da gioco (negli orari in cui questo non è occupato per le lezioni). A tal fine, è di interesse conoscere anche il costo per la prenotazione di ogni campo da gioco da gioco.</p>
--	--

2.2.3 Raffinamento delle Specifiche e Individuazione dei Concetti di Base

	FRASI DI CARATTERE GENERALE
	Non ce ne sono.

	FRASI RELATIVE ALLE PERSONE
	<p>Una persona è caratterizzata da un : nome, cognome , data di nascita , luogo di nascita e sesso.</p> <p>Una persona può prenotare un campo da gioco (negli orari in cui questo non è occupato per le lezioni). A tal fine, è di interesse conoscere anche il costo per la prenotazione di ogni campo da gioco da gioco.</p>

	FRASI RELATIVE AI MEDICI
	<p>caratterizzato da nome, cognome, sesso, luogo e data di nascita.</p> <p>Un medico è caratterizzato inoltre dalla data di conseguimento della abilitazione alla professione e dalla data di conseguimento della laurea</p> <p>visita medica effettuata da uno dei medici del centro, sia al momento della sottoscrizione che del rinnovo dell'abbonamento.</p>

	FRASI RELATIVE AI CLIENTI
	<p>Un cliente è caratterizzato da nome, cognome, sesso, luogo e data di nascita</p> <p>Un cliente può sottoscrivere un abbonamento che fornisce l'iscrizione ad una o più attività sportive.</p> <p>Un cliente deve sottoporsi ad una visita medica per il rilascio del certificato al momento della sottoscrizione e poi ad ogni rinnovo dell'abbonamento.</p>

	FRASI RELATIVE AGLI SPAZI
	<p>Il centro di fitness si compone di un certo numero di sale, di diversa tipologia (sala fitness, sala corsi), da uno o più spogliatoi maschili e femminili e da un certo numero di campi da gioco da gioco.</p> <p>Per sale e per campi da gioco, si vuole memorizzare il nome, univoco all'interno del fitness center, una descrizione e la capienza (espressa in numero di persone). Per ogni lezione, si vuole memorizzare la sala o il campo da gioco in cui si svolge e l'istruttore che la supervisiona.</p> <p>Una Persona, inoltre, può prenotare un campo da gioco (negli orari in cui questo non è occupato per le lezioni). A tal fine, è di interesse conoscere anche il costo per la prenotazione di ogni campo da gioco da gioco.</p>

	FRASI RELATIVE ALLE ATTIVITÀ SPORTIVE
	<p>Le attività sportive possono essere svolte in sala o in un campo da gioco. Il costo dell'abbonamento dipende dal costo delle singole attività sportive scelte dal cliente.</p> <p>Un'attività sportiva è caratterizzata da un nome univoco e da una descrizione</p> <p>Un'attività sportiva è caratterizzata inoltre da un numero massimo di partecipanti e da un numero di iscritti</p> <p>Per le attività sportive sono previste delle lezioni una o più lezioni a cadenza settimanale</p>

	FRASI RELATIVE ALLE LEZIONI
	<p>Le lezioni sono caratterizzate da un giorno, un orario di inizio ed una durata espressa in minuti</p> <p>Le lezioni inoltre hanno cadenza settimanale e sono relative alle attività sportive</p> <p>Una lezione si svolge in uno spazio che può essere un campo da gioco o una sala e viene supervisionata da un istruttore.</p>

	FRASI RELATIVE AGLI ABBONAMENTI
	<p>Un abbonamento è caratterizzato da una data di inizio e da una durata di validità ed fornisce l'iscrizione del cliente a una o più attività sportive. Alla scadenza dell'abbonamento bisogna creare uno nuovo , si può creare un abbonamento con attività sportive diverse da quelle inserite in quello precedente.</p>

	FRASI RELATIVE ALLE VISITE MEDICHE
	<p>Il cliente deve sottoporsi a una visita medica, effettuata da uno dei medici del centro, sia al momento della sottoscrizione che del rinnovo dell'abbonamento.</p> <p>Una visita medica è valida per la partecipazione a qualsiasi tipologia di attività sportiva.</p>

	FRASI RELATIVE AI BREVETTI
	<p>Un brevetto è identificato da un titolo, da un livello, una descrizione, è emesso da un ente sportivo e dà una qualifica per l'insegnamento di un determinato tipo di attività sportiva (nuoto, tennis, body-building...).</p> <p>Si vuole conoscere tutti i brevetti posseduti da un istruttore e la data di conseguimento degli stessi</p> <p>Un istruttore può avere conseguito uno o più brevetti.</p>

	FRASI RELATIVE AGLI ISTRUTTORI
	<p>Un istruttore deve avere o il titolo di diploma ISEF o di laurea in Scienze Motorie può avere conseguito uno o più brevetti.</p> <p>Un istruttore è assegnato alla supervisione di un certo numero di lezioni.</p>

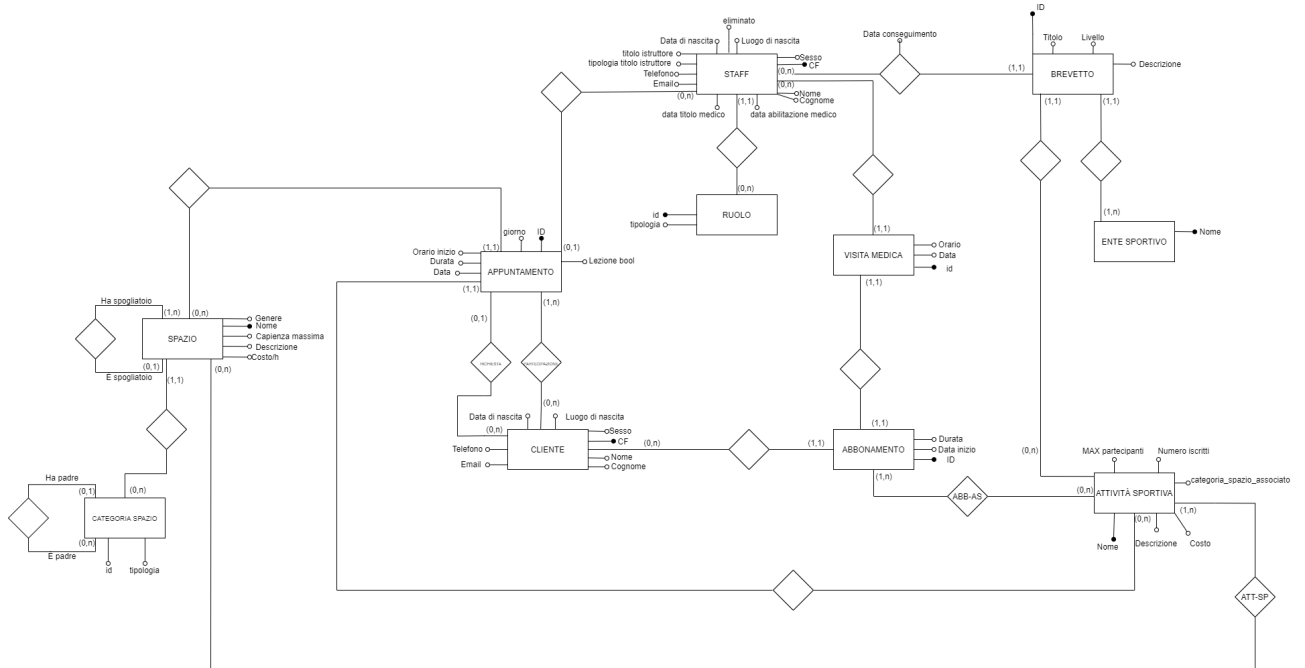
2.3 Specifica delle Operazioni sui dati previste

- [1] Inserire i dati relativi ad un nuovo cliente.
- [2] Modificare l'orario di una lezione.
- [3] Eliminare i dati relativi ad un determinato medico.
- [4] Visualizzare tutti le lezioni tenute da un determinato istruttore in un dato giorno della settimana.
- [5] Visualizzare il numero di abbonamenti sottoscritti da ciascun cliente.
- [6] Visualizzare il cliente che ha sottoscritto l'abbonamento con il numero massimo di corsi.
- [7] Visualizzare il cliente che ha prenotato lo stesso **campo da gioco** due volte in una settimana.
- [8] Visualizzare la sala in cui si svolge il minor numero di attività sportive.
- [9] Visualizzare gli istruttori che hanno solo il brevetto di nuoto.
- [10] Visualizzare i medici che sono anche clienti del centro ma non seguono attività che si svolgono in un dato orario.
- [11] Controllare che le lezioni per attività sportive svolte in sala abbiano come struttura di svolgimento la sala e idem per il **campo da gioco**.
- [12] Snellire le prenotazioni spostando le vecchie di oltre un anno in una entità separata (PRENOTAZIONI_STORICHE). Fare questa operazione ogni fine anno.

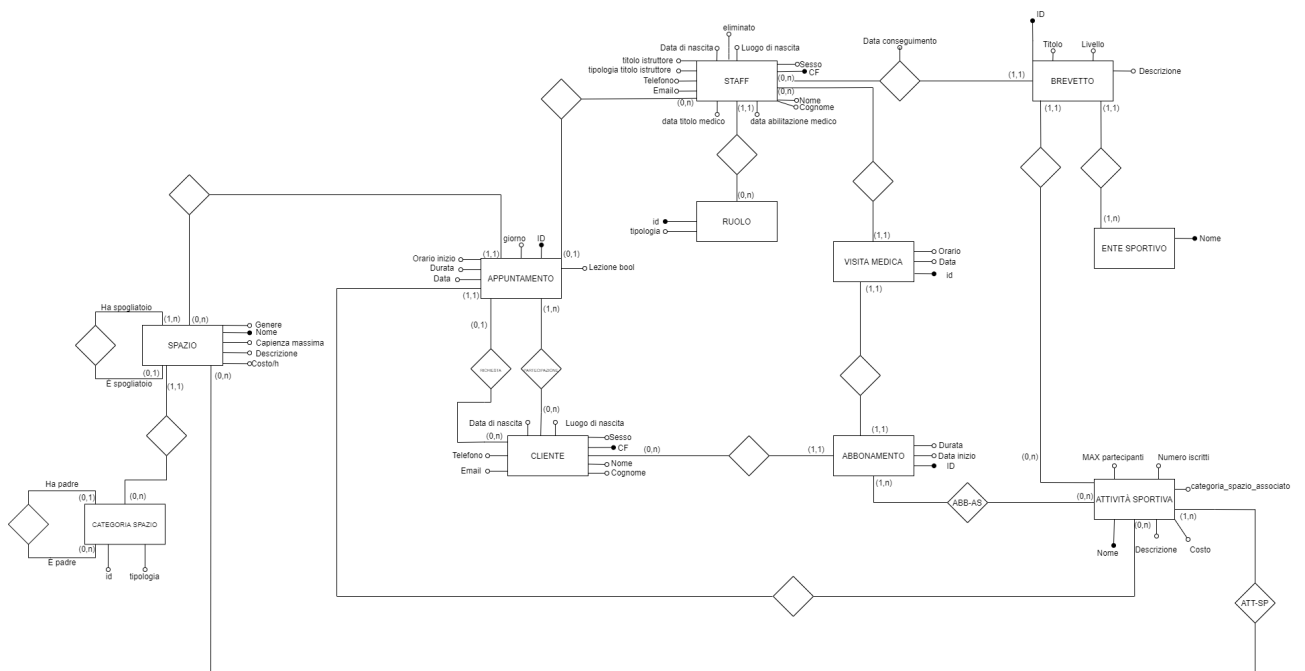
3 Progettazione Concettuale

3.1 Schemi E-R

Schema E-R



Schema E-R ristrutturato



3.2 Documentazione dello Schema E-R

3.2.1 Dizionario dei Dati

ENTITÀ	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORE
Cliente	clienti del fitness club che possono essere anche membri dello staff	CF, cognome, nome, sesso, data di nascita, luogo di nascita, telefono, email	CF
Staff	Comprende i dipendenti del fitness club quali i medici, il personale amministrativo e gli istruttori	CF, cognome, nome, sesso, data di nascita, luogo di nascita, telefono, email, titolo istruttore, data titolo medico, data abilitazione medico, tipologia titolo istruttore, eliminato	CF
Ruolo	definisce qual è il ruolo di una persona presente nello staff (medico personale amministrativo, istruttore o eventuali categorie inserite in un secondo momento)	id, tipologia	id
Spazio	Strutture e spazi posseduti, o comunque a disposizione del fitness club necessari per lo svolgimento degli incontri.	Nome, descrizione, capienza massima, genere, costo/h	Nome
Categoria Spazio	Definisce a quale categoria appartiene uno spazio, la quale può essere uno spogliatoio, un'area sportiva (e le sue sotto - categorie) o eventuali categorie introdotte in un momento successivo	id, tipologia	id
Attività Sportiva	Descrive le varie tipologie di attività sportiva che si possono svolgere nel fitness club	Nome, descrizione, costo, numero iscritti, max partecipanti, categoria_spazio_associato	Nome
Abbonamento	Consente di gestire gli abbonamenti sottoscritti dai clienti	ID, data inizio, durata	id
Visita Medica	Visita effettuata da un medico del centro ad un cliente alla sottoscrizione o al rinnovo dell'abbonamento	id, orario, data	id
Ente sportivo	Ente addetto al rilascio dei brevetti	Nome	Nome

Brevetto	Serve a tener traccia dei brevetti posseduti dagli istruttori	ID, titolo, livello, descrizione	ID
Appuntamento	Consente la gestione degli appuntamenti (lezioni, o prenotazioni dei campi da gioco)	ID, giorno, data, durata, orario di inizio, lezione bool	ID

3.2.2 Descrizione Entità

STAFF			
Descrizione	Entità che comprende i dipendenti del fitness club: il personale amministrativo, i medici e gli istruttori		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
CF	Alfanumerico	identificatore univoco	codice fiscale
Sesso	boolean	obbligatorio	true = maschio false = femmina
Nome	Alfanumerico	obbligatorio	nome di una persona
Cognome	Alfanumerico	obbligatorio	cognome di una persona
Data di nascita	data	obbligatorio	data di nascita di una persona
Luogo di nascita	Alfanumerico	obbligatorio	luogo di nascita di una persona
Telefono	Alfanumerico	opzionale	numero di telefono di una persona
Email	Alfanumerico	opzionale	e-mail di una persona

data titolo medico	data	opzionale	data del rilascio del titolo di un determinato medico
data abilitazione medico	data	opzionale	data del rilascio dell'abilitazione di un determinato medico
titolo istruttore	Alfanumerico	opzionale	titolo di studio di un istruttore
tipologia titolo istruttore	booleano	opzionale	true = laurea false = diploma isef
eliminato	booleano	opzionale	true se il membro dello staff è eliminato, false altrimenti

RUOLO			
Descrizione	Entità che definisce le possibili specializzazioni dello staff.		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
ID	contatore	identificatore univoco	numero incrementale, definisce univocamente un record
tipologia	Alfanumerico	obbligatorio	definisce la specializzazione e di una persona appartenente allo staff, le possibili specializzazioni sono : medico, personale amministrativo e istruttore.

APPUNTAMENTO			
Descrizione	Entità che raggruppa i vari tipi di incontri di cui può essere fatta richiesta, raggruppa sia le lezioni che le prenotazioni.		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
ID	contatore	identificatore univoco	numero incrementale che identifica univocamente un determinato appuntamento.
giorno	Alfanumerico	obbligatorio	giorno della settimana in cui si svolge una lezione
Ora inizio	Alfanumerico	obbligatorio	orario di inizio di una lezione o orario di una prenotazione
durata	Alfanumerico	obbligatorio	durata della lezione o della prenotazione espressa in minuti
Data	data	opzionale	data per la quale è prevista una prenotazione o una lezione
Lezione	boolean	obbligatorio	true = lezione false = prenotazione

SPAZIO			
Descrizione	Entità che generalizza i possibili spazi sia all'aperto (campetti) che al chiuso (sale e spogliatoi) presenti all'interno del fitness club.		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Nome	Alfanumerico	identificatore univoco	nome univoco dello spazio
Capienza massima	numerico	obbligatorio	numero massimo di persone che possono essere contemporaneamente all'interno dello stesso spazio.
Descrizione	Alfanumerico	opzionale	descrive lo spazio in questione
Costo/h	numerico	opzionale	costo orario di un campo
Genere	booleano	opzionale	true= spogliatoio maschile false= spogliatoio femminile

CATEGORIA SPAZIO			
Descrizione	Entità che generalizza i possibili spazi sia all'aperto (campetti) che al chiuso (sale e spogliatoi) presenti all'interno del fitness club.		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
ID	contatore	identificatore univoco	definisce univocamente il record

tipologia	Alfanumerico	obbligatorio	definisce la specializzazione di uno spazio appartenente al fitness club, le possibili specializzazioni sono : campo, sala e spogliatoio.
-----------	--------------	--------------	---

BREVETTO			
Descrizione	Entità che generalizza i possibili spazi sia all'aperto (campetti) che al chiuso (sale e spogliatoi) presenti all'interno del fitness club.		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
ID	numerico	obbligatorio	identificatore univoco.
titolo	Alfanumerico	obbligatorio	titolo rilasciato dal brevetto.
livello	Alfanumerico	obbligatorio	livello del brevetto conseguito
descrizione	Alfanumerico	obbligatorio	descrizione del brevetto

ENTE SPORTIVO			
Descrizione	Entità che generalizza i possibili spazi sia all'aperto (campetti) che al chiuso (sale e spogliatoi) presenti all'interno del fitness club.		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Nome	Alfanumerico	identificatore univoco	nome univoco dell'ente sportivo che rilascia un brevetto.

CLIENTE			
Descrizione	Un'entità che definisce un cliente del centro fitness , che può essere un cliente abbonato o non.		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
CF	Alfanumerico	identificatore univoco	Identifica univocamente un cliente
nome	Alfanumerico	obbligatorio	nome cliente
cognome	alfanumerico	obbligatorio	cognome cliente
data di nascita	data	obbligatorio	definisce la data in cui avviene l'appuntamento
telefono	Alfanumerico	opzionale	numero di telefono del cliente
email	Alfanumerico	opzionale	email del cliente
luogo di nascita	Alfanumerico	obbligatorio	definisce il luogo di nascita del cliente
sex	booleano	obbligatorio	definisce il sesso del cliente: true = maschio false = femmina

ABBONAMENTO			
Descrizione	Un'entità che definisce un abbonamento di un dato cliente.		
Nome Attributo	Tipo di dato	Vincolo	Descrizione

ID	contatore	identificatore univoco	Identifica univocamente un abbonamento
Durata	numerico	obbligatorio	definisce la durata dell'abbonamento
Data inizio	data	obbligatorio	definisce il giorno di inizio dell'abbonamento

VISITA MEDICA			
Descrizione	Un'entità che definisce una visita medica di un dato cliente.		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
ID	contatore	obbligatorio	identificatore univoco
Orario	Alfanumerico	obbligatorio	definisce l'orario della visita medica
Data	data	obbligatorio	definisce la data della visita medica

ATTIVITÀ SPORTIVA			
Descrizione	Un'entità che definisce un'attività sportiva disponibile nel centro fitness		
Nome Attributo	Tipo di dato	Vincolo	Descrizione
Nome	Alfanumerico	identificatore univoco	identifica univocamente

			un'attività sportiva
descrizione	Alfanumerico	obbligatorio	definisce una descrizione dell'attività sportiva
Max partecipanti	numerico	obbligatorio	definisce il numero massimo (in termini di persone) dell'attività sportiva.
numero iscritti	numerico	obbligatorio	numero di clienti iscritti ad una specifica attività sportiva.
costo	numerico	obbligatorio	costo dell'attività sportiva in relazione alla durata dell'abbonamento.
categoria_s pazio_associato	booleano	obbligatorio	indica se l'attività sportiva è da sala o da campo

3.2.3 Descrizione Relazioni

RELAZIONE: STAFF , APPUNTAMENTO	
DESCRIZIONE	UN ISTRUTTORE DELLO STAFF PUÒ DIRIGERE UNA LEZIONE
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
STAFF	(0,N)
APPUNTAMENTO	(1,1)

RELAZIONE: STAFF, RUOLO	
DESCRIZIONE	AD OGNI PERSONA DELLO STAFF È ASSOCIATO UN RUOLO
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
STAFF	(1,1)
RUOLO	(0,N)

RELAZIONE: STAFF, VISITA_MEDICA

DESCRIZIONE	UN MEDICO DELLO STAFF PUÒ PRESIDERE DELLE VISITE MEDICHE
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
STAFF	(0,N)
VISITA MEDICA	(1,1)

RELAZIONE: STAFF, BREVETTO		
DESCRIZIONE	OGNI ISTRUTTORE PUÒ AVERE UN CERTO NUMERO DI BREVETTI	
Entità Coinvolte		
ENTITÀ	CARDINALITÀ	
STAFF	(0,N)	
BREVETTO	(1,1)	
ATTRIBUTI		
NOME	TIPO DI DATI	DESCRIZIONE
DATA CONSEGUIMENTO	DATA	DATA IN CUI È STATO RILASCIATO IL BREVETTO

RELAZIONE: BREVETTO, ENTE SPORTIVO	
DESCRIZIONE	OGNI BREVETTO È RILASCIATO DA UN DETERMINATO ENTE SPORTIVO
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
BREVETTO	(1,1)
ENTE SPORTIVO	(1,N)

RELAZIONE: BREVETTO, ATTIVITÀ SPORTIVA	
DESCRIZIONE	OGNI BREVETTO È RELATIVO AD UNA SPECIFICA ATTIVITÀ SPORTIVA
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
BREVETTO	(1,1)
ATTIVITÀ SPORTIVA	(0,N)

RELAZIONE: ABBONAMENTO, VISITA MEDICA	
DESCRIZIONE	ALLA SOTTOSCRIZIONE DELL'ABBONAMENTO E POI AD OGNI RINNOVO DEVE ESSERE FATTA UNA VISITA MEDICA

Entità Coinvolte	
ENTITÀ	CARDINALITÀ
ABBONAMENTO	(1,1)
VISITA MEDICA	(1,1)

RELAZIONE: APPUNTAMENTO, SPAZIO	
DESCRIZIONE	AD OGNI LEZIONE O PRENOTAZIONE È ASSOCIATO UNO SPAZIO IN CUI ESSA SI SVOLGE
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
APPUNTAMENTO	(1,1)
SPAZIO	(0,N)

RELAZIONE: SPAZIO, SPAZIO	
DESCRIZIONE	OGNI SPAZIO PUÒ O AVERE UNO SPOGLIATOIO O ESSERE UNO SPOGLIATOIO
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
SPAZIO (HA SPOGLIATOIO)	(1,N)
SPAZIO(È SPOGLIATOIO)	(0,1)

RELAZIONE : SPAZIO , CATEGORIA SPAZIO	
DESCRIZIONE	LA RELAZIONE DEFINISCE UN'ASSOCIAZIONE TRA UNO SPAZIO ED UNA CATEGORIA
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
SPAZIO	(1,1)
CATEGORIA SPAZIO	(0,N)

RELAZIONE : PARTECIPAZIONE	
DESCRIZIONE	DEFINISCE UN'ASSOCIAZIONE DOVE OGNI APPUNTAMENTO HA ALMENO UN CLIENTE
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
APPUNTAMENTO	(1,N)
CLIENTE	(0,N)

RELAZIONE : CLIENTE , ABBONAMENTO	
DESCRIZIONE	DEFINISCE UN'ASSOCIAZIONE TRA CLIENTE ED ABBONAMENTO
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
CLIENTE	(0,N)
ABBONAMENTO	(1,1)

RELAZIONE : ABB-AS	
DESCRIZIONE	DEFINISCE UNA COPPIA DATA DA UN ABBONAMENTO E UN'ATTIVITÀ SPORTIVA
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
ATTIVITÀ SPORTIVA	(0,N)
ABBONAMENTO	(1,N)

RELAZIONE : ATT-SP	
DESCRIZIONE	DEFINISCE UNA COPPIA DATA DA UN'ATTIVITÀ SPORTIVA E SPAZIO
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
ATTIVITÀ SPORTIVA	(1,N)
SPAZIO	(0,N)

RELAZIONE : RICHIESTA	
DESCRIZIONE	UN CLIENTE PUÒ PRENOTARE DEI CAMPI DA GIOCO ED UNA PERNOTAZIONE È EFFETTUATA DA UN SOLO CLIENTE
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
CLIENTE	(0,N)
APPUNTAMENTO	(0,1)

RELAZIONE : CATEGORIA SPAZIO , CATEGORIA SPAZIO	
DESCRIZIONE	UNA CATEGORIA SPAZIO PUÒ RIFERIRSI AD UNA CATEGORIA SPAZIO DI LIVELLO SUPERIORE
Entità Coinvolte	
ENTITÀ	CARDINALITÀ
HA PADRE	(0,1)
È PADRE	(0,N)

3.2.4 Vincoli non espressi dallo schema E/R

Vincoli non espressi dallo schema E-R

Tabella 1.

Nota: Il numero riportato alla sinistra delle regole di vincolo rappresenta l'ordine in cui appaiono nelle tabelle che contengono il codice di definizione dei trigger

PROGR ESSIVO	REGOLE DI VINCOLO
6	Un cliente può prenotare un campo da gioco, ma non una sala
19	Un cliente può partecipare solo alle lezioni di attività sportive che rientrano nel suo abbonamento in corso di validità
9	Un istruttore non può supervisionare lezioni la cui attività sportiva non rientra nei suoi brevetti
12	Uno spogliatoio deve avere un'area sportiva associata, ma non uno spogliatoio e un'area sportiva non può essere spogliatoio di altri spazi
8	Gli spogliatoi devono avere esattamente un genere, ma le aree sportive no
7	Solo i campi da gioco devono avere un costo orario associato
13	In un area sportiva non possono svolgersi appuntamenti relativi ad attività sportive per le quali non è adatta
14	Gli orari degli appuntamenti non devono sovrapporsi
1	Solo un medico può effettuare delle visite mediche
3	Solo un medico ha una data di conseguimento titolo ed una data di conseguimento abilitazione
2	Solo un istruttore ha un titolo istruttore ed una tipologia titolo istruttore
4	Un cliente deve rilasciare una mail o un numero di telefono , ma non obbligatoriamente entrambi
5	Un membro dello staff deve rilasciare una mail o un numero di telefono , ma non obbligatoriamente entrambi
10	Solo un istruttore può avere dei brevetti
18	Il numero di partecipanti ad una lezione non deve eccedere il numero la capienza massima dell'area sportiva
15	Il numero di iscritti ad una attività sportiva deve essere minore della capacità massima
17	Solo le aree sportive possono avere attività sportive associate
20	Uno spazio non può avere come id_categoria_spazio un id relativo a 'sala' o 'campo' , ma solo ai figli oppure ad uno spogliatoio
21	Un appuntamento che è una lezione deve avere obbligatoriamente un istruttore che la supervisiona.
22	Un abbonamento non può avere associata una visita medica con data superiore alla data dell'abbonamento stesso

23	Un appuntamento che è una lezione non deve avere associato un cliente che prenota , se invece è una prenotazione deve avere obbligatoriamente un cliente che prenota
24	Un appuntamento può avere come giorno un valore compreso in questo insieme ={lunedì,martedì,mercoledì,giovedì,venerdì,sabato,domenica}

Altri vincoli

	Integrità dati ed altro
11	il numero iscritti in attività sportiva (essendo un dato ridondato, va calcolato come numero di clienti che hanno un abbonamento in corso di validità per quella attività sportiva) deve essere minore della capacità massima
16	Aggiornamento del numero di partecipanti all'inserimento di un'attività sportiva tra quelle di un abbonamento
25	Aggiornamento del numero di iscritti alle attività sportive allo scadere degli abbonamenti

4 Progettazione Logica (Traduzione dello Schema E-R)

4.1 Traduzione Entità

4.1.1 Descrizione Relazioni

attività_sportive			
Descrizione	Attività sportive che si svolgono nel centro fitness		
Nome Attributo	Tipo di Dato	Vincolo	Descrizione
nome	varchar(50)	pk	nome con cui comunemente si indica l'attività sportiva (es nuoto, staffetta)

descrizione	varchar(1000)		descrizione degli esercizi che si svolgono durante l'attività
costo	decimal		costo orario che è valorizzato solo per i campi da gioco
numero_iscritti	integer	default 0	numero di clienti che hanno un abbonamento in corso di validità per quella attività sportiva
max_partecipanti	integer	not null	numero massimo di clienti che possono iscriversi all'attività sportiva
categoria_spazio_associato	boolean	not null	0 se l'attività sportiva è da sala, 1 se l'attività sportiva è da campo

appuntamenti			
Descrizione	Definisce un appuntamento avvenuto nel fitness club		
Nome Attributo	Tipo di Dato	Vincolo	Descrizione
id	integer	pk, auto_increment	identificatore univoco di un appuntamento
giorno	varchar(20)	not null	giorno in cui avviene l'appuntamento
orario_inizio	time	not null	l'orario in cui inizia l'appuntamento
durata	integer	not null	durata degli appuntamenti
data	date	not null	la data dell'appuntamento
lezione_bool	boolean	not null	definisce se è una lezione o un abbonamento : true = lezione , false = prenotazione.
nome_attivita_sportiva	varchar(50)	not null , fk	chiave esterna associata alla relazione attività_sportive
cf_istruttore	char(16)	fk	chiave esterna associata alla relazione staff
nome_spazio	varchar(50)	not null,fk	chiave esterna associata alla relazione spazio

cf_prenotazione	char(16)	fk	codice fiscale del cliente che effettua la prenotazione
-----------------	----------	----	---

brevetti			
Descrizione			
Nome Attributo	Tipo di Dato	Vincolo	Descrizione
Id	integer	PK,auto_incremento	Chiave primaria
cf_staff	char(16)	FK	chiave esterna associata alla relazione con staff
titolo	varchar(50)	not null	titolo conferito dal brevetto
livello	varchar(50)	not null	livello del brevetto
descrizione	varchar(1000)	not null	descrizione del brevetto
nome_ente_sportivo	varchar(20)	FK, not null	chiave esterna associata alla relazione con ente sportivo
nome_attivita_sportiva	varchar(50)	FK, not null	chiave esterna associata alla relazione con attività sportiva

enti_sportivi			
Descrizione			
Nome Attributo	Tipo di Dato	Vincolo	Descrizione
nome	varchar(20)	PK	nome univoco di un ente sportivo

ruoli			
Descrizione			
Nome Attributo	Tipo di Dato	Vincolo	Descrizione
id	integer	PK,auto_increment	id univoco che si riferisce a un ruolo
tipologia	varchar(50)	not null	specializzazione di staff

visite_mediche			
Descrizione	visite mediche a cui i clienti devono sottoporsi alla scadenza ed al rinnovo degli abbonamenti		
Nome Attributo	Tipo di Dato	Vincolo	Descrizione
ID	integer	PK, auto_increment	id della visita medica
orario	time	not null	orario di svolgimento della visita
data	date	not null	data nella quale si svolge la visita
CF_staff	char(16)	FK, not null	codice fiscale del medico che sovrintende la visita

abbonamenti			
Descrizione	Abbonamenti sottoscritti dai clienti		
Nome Attributo	Tipo di Dato	Vincolo	Descrizione
id	integer	PK,auto_increment	chiave primaria di abbonamento
durata	integer	not null	durata in giorni dell'abbonamento
data_inizio	date	not null	data dalla quale inizia il periodo di validità dell'abbonamento
cf_cliente	char(16)	FK, not null	cf del cliente a cui è intestato l'abbonamento
id_visita_medica	integer	FK, not null	visita medica associata all'abbonamento

abbonamenti_attivita_sportive			
Descrizione			
Nome Attributo	Tipo di Dato	Vincolo	Descrizione
ID_abbonamento	integer	PK, FK, auto_increment	chiave primaria di abbonamento
nome_attivita_sportiva	varchar(50)	PK, FK	nome univoco dell'attività sportiva a cui fa riferimento un abbonamento

clienti			
Descrizione	clienti del fitness club (non per forza abbonati)		
Nome Attributo	Tipo di Dato	Vincolo	Descrizione
cf	char(16)	PK	chiave primaria di cliente
cognome	varchar(30)	not null	cognome cliente
nome	varchar(30)	not null	nome cliente
telefono	varchar(15)		numero di telefono cliente
email	varchar(50)		email cliente
luogo_di_nascita	varchar(30)	not null	luogo di nascita cliente
data_di_nascita	date	not null	data di nascita cliente
sesto	boolean	not null	data di nascita del cliente

staff			
Descrizione	personale (dipendenti) del fitness club		
Nome Attributo	Tipo di Dato	Vincolo	Descrizione
cf	char(16)	PK	chiave primaria staff
cognome	varchar(30)	not null	cognome staff
nome	varchar(30)	not null	nome staff
telefono	varchar(15)		numero di telefono staff
email	varchar(50)		email staff
luogo_di_nascita	varchar(30)	not null	luogo di nascita staff
data_di_nascita	date	not null	data di nascita staff
sesto	boolean	not null	sesto staff
titolo_istruttore	varchar(50)		titolo istruttore
tipologia_titolo_istruttore	boolean		tipologia del titolo istruttore
data_titolo_medico	date		data conseguimento laurea medico
data_abilitazione_medico	date		data di abilitazione medico
id_ruolo	integer	not null	ruolo della persona appartenente allo staff
eliminato	boolean		indica se il membro dello staff è stato eliminato

categoria_spazi			
Descrizione	la categoria di ogni spazio		
Nome Attributo	Tipo di Dato	Vincolo	Descrizione
id	integer	pk ,auto_increment	chiave primaria e auto_increment e definisce in modo univoco una categoria
tipologia	varchar(50)	not null	definisce con una descrizione significativa la categoria di spazio
tipologia_padre	integer	fk	chiave esterna ricorsiva

spazi			
Descrizione	Spazi a disposizione del fitness club in cui si svolgono gli appuntamenti		
Nome Attributo	Tipo di Dato	Vincolo	Descrizione
nome	varchar(50)	pk	nome e chiave primaria degli spazi in cui si svolgono gli appuntamenti
descrizione	varchar(1000)		descrizione dello spazio (descrizione del luogo in cui si trova)
capienza_massima	integer	not null	numero massimo di partecipanti che lo spazio può accogliere
genere	boolean		genere associato allo spazio (sarà effettivamente associato solo agli spogliatoi)
costo_h	decimal		costo orario associato ai campi da gioco
id_categoria_spazio	integer	fk, not null	indica a quale categoria fa parte lo spazio

area_associata	varchar(50)	fk	indica lo spogliatoio associato allo spazio
----------------	-------------	----	---

attività_sportive_spazi			
Descrizione	Definisce una relazione tra Attività sportiva e spazio		
Nome Attributo	Tipo di Dato	Vincolo	Descrizione
nome_attività_sportiva	varchar(50)	PK, FK	chiave primaria-chiave secondaria(clienti)
nome_spazio	varchar(50)	PK, FK	chiave primaria-chiave secondaria(spazi)

partecipazioni			
Descrizione	Definisce una relazione tra un appuntamento e un cliente		
Nome Attributo	Tipo di Dato	Vincolo	Descrizione
id	integer	fk , pk	chiave primaria combinata , chiave esterna associata ad appuntamento
cf	char(16)	fk,pk	chiave primaria combinata, chiave esterna associata a cliente

RELAZIONI ULTERIORI NON DERIVATE DALLO SCHEMA CONCETTUALE

La base di dati contiene un'altra relazione, non descritta a livello concettuale perché esula dalla realtà di interesse. È però necessario, vista la richiesta numero 12, creare questa relazione che contiene le prenotazioni storiche. La relazione non ha vincoli di integrità referenziale con altre relazioni perché i dati sono presi dalla tabella appuntamenti dove i vincoli ci sono ed i dati vengono inseriti solo dalla stored routine definita per questo scopo. Inoltre la tabella si può rendere non modificabile da nessun utente ed i dati verranno solo utilizzati per indagini statistiche. Prenotazioni storiche è quindi da considerare più un archivio che una relazione, al pari delle altre, nella base di dati.

Schema della relazione

prenotazioni_storiche			
Descrizione	Definisce un appuntamento avvenuto nel fitness club		
Nome Attributo	Tipo di Dato	Vincolo	Descrizione
id	integer	pk	identificatore univoco della prenotazione
giorno	varchar(20)	not null	giorno in cui avviene l'incontro

orario_inizio	time	not null	l'orario in cui inizia l'incontro
durata	integer	not null	durata dell'incontro
data	date	not null	la data in cui si tiene l'incontro
nome_attivita_sportiva	varchar(50)	not null	chiave esterna associata alla relazione attività_sportive
cf_istruttore	char(16)		chiave esterna associata alla relazione staff
nome_spazio	varchar(50)	not null	chiave esterna associata alla relazione spazio

5 Progettazione Fisica

5.1 Definizione dello schema della base di dati

(Script SQL-DDL)

Definizione delle relazioni

attività_sportive
<pre>create table attività_sportive(nome varchar(50), descrizione varchar(1000), costo decimal, numero_iscritti integer default 0, max_partecipanti integer not null, categoria_spazio_associato boolean not null, primary key(nome));</pre>

categoria_spazi
<pre>create table categoria_spazi(id integer auto_increment, tipologia varchar(50) not null,</pre>

```
tipologia_padre integer,  
  
primary key(id),  
foreign key (tipologia_padre) references categoria_spazi(id));
```

spazi

```
create table spazi(  
nome varchar(50),  
descrizione varchar(1000),  
capienza_massima integer not null,  
genere boolean,  
costo_h decimal,  
id_categoria_spazio integer not null,  
area_associata varchar(50),  
  
primary key(nome),  
foreign key(id_categoria_spazio) references categoria_spazi(id),  
foreign key(area_associata) references spazi(nome));
```

clienti

```
create table clienti(  
cf char(16),  
cognome varchar(30) not null,  
nome varchar(30) not null,  
telefono varchar(15),  
email varchar(50),  
luogo_di_nascita varchar(50) not null,  
data_di_nascita date not null,  
sesso boolean not null,  
  
primary key(cf));
```

staff

```
create table staff(  
cf char(16),  
cognome varchar(30) not null,
```

```
nome varchar(30) not null,  
telefono varchar(15),  
email varchar(50),  
luogo_di_nascita varchar(30) not null,  
data_di_nascita date not null,  
sesso boolean not null,  
titolo_istruttore varchar(50),  
tipologia_titolo_istruttore boolean,  
data_titolo_medico date,  
data_abilitazione_medico date,  
id_ruolo integer not null,  
eliminato boolean default 0,  
  
primary key(cf),  
foreign key(id_ruolo) references ruoli(id));
```

ruoli

```
create table ruoli(  
id integer auto_increment,  
tipologia varchar(50) not null,  
  
primary key(id));
```

enti_sportivi

```
create table enti_sportivi(  
nome varchar(20),  
primary key(nome));
```

brevetti

```
create table brevetti(  
id integer auto_increment,  
cf_staff char(16) not null,  
titolo varchar(50) not null,  
livello integer not null,  
descrizione varchar(1000),
```



```
data_conseguimento date not null,  
nome_ente_sportivo varchar(20) not null,  
nome_attivita_sportiva varchar(50) not null,  
  
primary key(id),  
foreign key(cf_staff) references staff(cf),  
foreign key(nome_ente_sportivo) references enti_sportivi(nome),  
foreign key(nome_attivita_sportiva) references attivita_sportive (nome));
```

appuntamenti

```
create table appuntamenti(  
id integer auto_increment,  
giorno varchar(20) not null,  
orario_inizio time not null,  
durata integer not null,  
data date not null,  
lezione_bool boolean not null,  
nome_attivita_sportiva varchar(50) not null,  
cf_istruttore char(16),  
nome_spazio varchar(50) not null,  
cf_prenotazione char(16),  
  
primary key(id),  
foreign key(nome_attivita_sportiva) references attivita_sportive (nome),  
foreign key(cf_istruttore) references staff(cf),  
foreign key(nome_spazio) references spazi(nome),  
foreign key(cf_prenotazione) references clienti(cf));
```

abbonamenti

```
create table abbonamenti(  
id integer auto_increment,
```

```
durata integer not null,  
data_inizio date not null,  
cf_cliente char(16) not null,  
id_visita_medica integer not null,  
  
primary key(id),  
foreign key (cf_cliente) references clienti(cf),  
foreign key(id_visita_medica) references visite_mediche(id));
```

visite_mediche

```
create table visite_mediche(  
id integer auto_increment,  
cf_staff char(16),  
orario time not null,  
data date not null,  
  
primary key(id),  
foreign key(cf_staff) references staff(cf));
```

attivita_sportive_spazi

```
create table attivita_sportive_spazi(  
nome_attivita_sportive varchar(50),  
nome_spazio varchar(50),  
  
primary key(nome_attivita_sportive, nome_spazio),  
foreign key(nome_attivita_sportive) references attivita_sportive(nome),  
foreign key(nome_spazio) references spazi(nome));
```

partecipazioni

```
create table partecipazioni(  
id_appuntamento integer,  
cf_cliente char(16),  
  
primary key (id_appuntamento ,cf_cliente),  
foreign key(id_appuntamento) references appuntamenti(id),
```

```
foreign key(cf_cliente) references clienti(cf));
```

abbonamenti_attivita_sportive

```
create table abbonamenti_attivita_sportive(  
id_abbonamento integer,  
nome_attivita_sportiva varchar(50),  
  
primary key(id_abbonamento, nome_attivita_sportiva),  
foreign key(nome_attivita_sportiva) references attivita_sportive(nome),  
foreign key(id_abbonamento) references abbonamenti(id));
```

Relazioni ulteriori

prenotazioni_storiche

```
create table prenotazioni_storiche(  
id integer,  
giorno varchar(20) not null,  
orario_inizio time not null,  
durata integer not null,  
data date not null,  
nome_attivita_sportiva varchar(50) not null,  
cf_istruttore char(16) not null,  
nome_spazio varchar(50) not null,  
  
primary key(id));
```

Triggers

5.2 Definizione dei trigger (aggiuntivi rispetto a quelli richiesti)

1	Solo un medico può effettuare delle visite mediche
	<pre>create or replace trigger medici_visite_mediche before insert on visite_mediche for each row begin /*solo un medico può effettuare una visita medica*/ declare id_medico integer; declare cf_medico char(16); declare elim boolean;</pre>

	<pre> select id, eliminato into id_medico, elim from ruoli where tipologia = 'medico'; if elim then signal sqlstate '45000' set message_text = 'Attenzione! Il medico selezionato è stato eliminato'; end if; select cf into cf_medico from staff where id_ruolo = id_medico && new.cf_staff=cf; if cf_medico is null then signal sqlstate '45000' set message_text = 'Attenzione! il codice fiscale inserito non corrisponde ad un medico'; end if; end// </pre>
--	--

2	Solo un istruttore ha un titolo istruttore ed una tipologia titolo istruttore
	<pre> create or replace trigger staff_istruttore before insert on staff for each row begin if new.id_ruolo in (select id from ruoli where tipologia = 'istruttore') then /*gli attributi relativi all'istruttore devono essere valorizzati*/ if((new.titolo_istruttore is null new.tipologia_titolo_istruttore is null)) then signal sqlstate '45000' set message_text = 'Attenzione! un istruttore deve avere un titolo istruttore ed una tipologia titolo valorizzati'; end if; /*gli attributi non relativi all'istruttore devono essere lasciati vuoti*/ if(new.data_titolo_medico is not null new.data_abilitazione_medico is not null) then signal sqlstate '45000' set message_text = 'Attenzione! Lasciare vuoti i campi data_titolo medico e data_abilitazione_medico'; end if; end if; end if; end// </pre>

3	Solo un medico ha una data di conseguimento titolo ed una data di conseguimento abilitazione
----------	---

	<pre> create or replace trigger staff_medici before insert on staff for each row begin declare id_medico integer; select id into id_medico from ruoli where tipologia = 'medico'; if new.id_ruolo = id_medico then if new.data_titolo_medico is null new.data_abilitazione_medico is null then signal sqlstate '45000' set message_text = 'Attenzione! Un medico deve avere una data di conseguimento laurea ed una di conseguimento abilitazione'; end if; if new.titolo_istruttore is not null new. tipologia_titolo_istruttore is not null then signal sqlstate '45000' set message_text = 'Attenzione! Lasciare vuoti i campi titolo_istruttore e tipologia_titolo_istruttore'; end if; end if; end if; end// </pre>
--	--

4	Un cliente deve rilasciare una mail o un numero di telefono , ma non obbligatoriamente entrambi
	<pre> create or replace trigger clienti_contatti before insert on clienti for each row begin if new.email is null && new.telefono is null then signal sqlstate '45000' set message_text = 'Attenzione! Almeno un recapito tra email e telefono deve essere inserito'; end if; end// </pre>

5	Un membro dello staff deve rilasciare una mail o un numero di telefono , ma non obbligatoriamente entrambi
----------	---

	<pre> create or replace trigger staff_contatti before insert on staff for each row begin /*Un membro dello staff deve rilasciare una mail o un numero di telefono , ma non obbligatoriamente entrambi*/ if new.email is null && new.telefono is null then signal sqlstate '45000' set message_text = 'Attenzione! Almeno un recapito tra email e telefono deve essere inserito'; end if; end// </pre>
--	---

6	UN CLIENTE PUÒ PRENOTARE UN CAMPO DA GIOCO, MA NON UNA SALA
	<pre> create or replace trigger prenotazione_cliente before insert on appuntamenti for each row begin /*contiene la categoria a cui appartiene lo spazio inserito*/ declare index_categoria integer; /*contiene la chiave associata alla categoria campo che è una di quelle predefinite*/ declare index_campo integer; select id_categoria_spazio into index_categoria from spazi where nome = new.nome_spazio; select id into index_campo from categoria_spazi cs where cs.tipologia = 'campo'; if index_categoria not in (select cs.id from categoria_spazi cs where cs.tipologia_padre = index_campo) and new.lezione_bool = false THEN signal sqlstate '45000' set message_text = 'Lo spazio associato deve essere un campo da gioco'; end if; end// </pre>

7	SOLO I CAMPI DA GIOCO DEVONO AVERE UN COSTO ORARIO ASSOCIATO
---	--

	<pre> create or replace trigger costo_campo before insert on spazi for each row BEGIN /*Solo i campi da gioco devono avere un costo orario associato*/ declare id_campo integer; select id into id_campo from categoria_spazi cs where cs.tipologia = 'campo'; if new.id_categoria_spazio IN (select id from categoria_spazi cs where cs.tipologia_padre = id_campo) then if new.costo_h IS NULL then signal sqlstate '45000' set message_text = 'Attenzione! Un campo deve avere anche un costo orario'; end if; else if new.costo_h IS NOT NULL then signal sqlstate '45000' set message_text = 'Attenzione! Solo i campo possono avere un costo orario associato'; end if; end if; END// </pre>
--	---

8	GLI SPOGLIATOI DEVONO AVERE ESATTAMENTE UN GENERE, MA LE AREE SPORTIVE NO
	<pre> create or replace trigger spazio_spgliatoio before insert on spazi for each row begin /*la categoria alla quale lo spazio che si sta inserendo appartiene*/ declare index_categoria_spazio integer; select id into index_categoria_spazio from categoria_spazi where tipologia = 'spogliatoio'; if new.id_categoria_spazio = index_categoria_spazio then if new.genere is null then signal sqlstate '45000' set message_text = 'Attenzione! Per uno spogliatoio deve esserci un genere associato'; </pre>

	<pre> end if; else if new.genere is not null then signal sqlstate '45000' set message_text = 'Attenzione! Solo gli spogliatoi devono avere un genere associato'; end if; end if; end// </pre>
--	---

9	<p>UN ISTRUTTORE NON PUÒ SUPERVISIONARE LEZIONI LA CUI ATTIVITÀ SPORTIVA NON RIENTRA NEI SUOI BREVETTI</p>
	<pre> create or replace trigger istruttore_appuntamento before insert on appuntamenti for each row begin /*Un istruttore non può supervisionare lezioni la cui attività sportiva non rientra nei suoi brevetti*/ /*contiene il nome dell'attività sportiva che si sta inserendo se l'istruttore ha il brevetto, altrimenti è null*/ declare attivita_brevetto varchar(50); /*indica se l'istruttore è stato eliminato*/ declare elim boolean; select nome_attivita_sportiva into attivita_brevetto from brevetti where cf_staff= new.cf_istruttore && new.nome_attivita_sportiva = nome_attivita_sportiva; select eliminato into elim from staff where cf = new.cf_istruttore; if elim then signal sqlstate '45000' set message_text = 'Attenzione! L\istruttore selezionato è stato eliminato'; end if; if attivita_brevetto is null then signal sqlstate '45000' set message_text = 'Attenzione! L\istruttore scelto non ha un brevetto per l\'attività sportiva selezionata'; end if; end// </pre>

--	--

10	SOLO UN ISTRUTTORE PUÒ AVERE DEI BREVETTI
	<pre> create or replace trigger istruttori_brevetti before insert on brevetti for each row begin declare id_ruolo_inserito integer; declare id_ruolo_istruttore integer; select id_ruolo into id_ruolo_inserito from staff where cf = new.cf_staff; select id into id_ruolo_istruttore from ruoli where tipologia = 'istruttore'; /*non occorre controllare che le variabili non siano nulle perché il vincolo di*/ /*integrità referenziale impone che la persona in staff esista ed il ruolo istruttore è sempre presente in ruoli*/ if id_ruolo_inserito != id_ruolo_istruttore then signal sqlstate '45000' set message_text = 'Attenzione! Solo un istruttore può conseguire dei brevetti'; end if; end// </pre>

12	UNO SPOGLIATOIO DEVE AVERE UN'AREA SPORTIVA ASSOCIATA, MA NON UNO SPOGLIATOIO E UN'AREA SPORTIVA NON PUÒ ESSERE SPOGLIATOIO DI ALTRI SPAZI
	<pre> create or replace trigger aree_sportive_spazi before insert on spazi for each row begin /*id associato alla categoria spogliatoi*/ declare index_spgliatoio integer; /*categoria dell'area associata allo spazio che stiamo inserendo*/ declare index_categoria_area_associata integer; select id into index_spgliatoio from categoria_spazi where tipologia = 'spogliatoio'; select id_categoria_spazio into index_categoria_area_associata from spazi where nome = new.area_associata; /*sto inserendo uno spogliatoio e può avere solo aree sportive associate*/ if new.id_categoria_spazio = index_spgliatoio </pre>

	<pre> then if new.area_associata is null then signal sqlstate '45000' set message_text = 'Attenzione! Uno spogliatoio deve avere un\'area sportiva associata'; end if; if index_categoria_area_associata = index_spogliatoio then signal sqlstate '45000' set message_text = 'Attenzione! Uno spogliatoio non può avere un altro spogliatoio associato'; end if; else /*non sto inserendo uno spogliatoio e quindi può avere solo uno spogliatoio associato*/ if new.area_associata is not null then signal sqlstate '45000' set message_text = 'Attenzione! Un area sportiva non può avere spazi associati'; end if; end if; end// </pre>
--	--

13	<p>IN UN AREA SPORTIVA NON POSSONO SVOLGERSI APPUNTAMENTI RELATIVI AD ATTIVITÀ SPORTIVE PER LE QUALI NON È ADATTA</p>
	<pre> create or replace trigger area_sportiva_attivita_sportiva before insert on appuntamenti for each row begin declare nome_attivita_sportiva_test varchar(100) ; select nome_attivita_sportiva into nome_attivita_sportiva_test from attivita_sportive_spazi where nome_attivita_sportiva = new.nome_attivita_sportiva and nome_spazio = new.nome_spazio ; if nome_attivita_sportiva_test is null then signal sqlstate '45000' set message_text = 'Attenzione! In un area sportiva non possono svolgersi appuntamenti relativi ad attività sportive per le quali non è adatta'; end if; end// </pre>

14	<p>GLI ORARI DEGLI APPUNTAMENTI NON DEVONO SOVRAPPORSI</p>
----	---

	<pre> create or replace trigger sovrapposizione_appuntamento before insert on appuntamenti for each row BEGIN /*la sovrapposizione si ha quando esiste già un appuntamento che comincia prima e finisce dopo l'ora d'inizio di quello nuovo oppure se ne inizia un altro prima dell'orario di fine del nuovo appuntamento*/ declare orario int default hour(new.orario_inizio)*60 + minute(new.orario_inizio); declare id_appuntamento integer; declare orario_fine int default hour(new.orario_inizio)*60 + minute(new.orario_inizio) + new.durata; select id into id_appuntamento from appuntamenti as a where new.data = data AND new.nome_spazio = nome_spazio AND (hour(a.orario_inizio)*60 + minute(a.orario_inizio) <= orario AND hour(a.orario_inizio)*60 + minute(a.orario_inizio) + durata >= orario) OR (hour(a.orario_inizio)*60 + minute(a.orario_inizio) <= orario_fine AND hour(a.orario_inizio)*60 + minute(a.orario_inizio) + durata >= orario_fine) ; if id_appuntamento IS NOT NULL then signal sqlstate '45000' set message_text = 'Due appuntamenti non possono essere fissati alla stessa ora per lo stesso campo'; end if; END// </pre>

15	Il numero di iscritti ad una attività sportiva deve essere zero al momento dell'inserimento
	<pre> create or replace trigger attivita_sportive_iscritti before insert on attivita_sportive for each row begin if new.numero_iscritti is not null then if new.numero_iscritti != 0 then signal sqlstate '45000' set message_text = 'Attenzione! Il numero di iscritti deve essere di zero unità'; end if; end if; end// </pre>

17	Solo le aree sportive possono avere attività sportive associate
	<pre> create or replace trigger attivita_sportiva_area_sportiva before insert on attivita_sportive_spazi for each row begin delimiter // declare index_spogliatoio integer ; declare categoria_spazio integer; select id into index_spogliatoio from categoria_spazi where tipologia = 'spogliatoio'; select id_categoria_spazio into categoria_spazio from spazi where new.nome_spazio = nome; if categoria_spazio = index_spogliatoio then signal sqlstate '45000' set message_text = 'Attenzione! Solo le aree sportive </pre>

	possono avere attività sportive associate' ; end if; end//
--	--

18	Il numero di partecipanti ad una lezione non deve eccedere il numero la capienza massima dell'area sportiva
	<pre> create or replace trigger lezione_partecipanti before insert on partecipazioni for each row begin /*Il numero di partecipanti ad una lezione non deve eccedere il numero la capienza massima dell'area sportiva*/ declare nome_area varchar(50); declare capienza_area integer; declare partecipanti_appuntamento integer; select nome_spazio into nome_area from appuntamenti where id = new.id_appuntamento; select capienza_massima into capienza_area from spazi where nome = nome_area; select count(*) into partecipanti_appuntamento from partecipazioni where id_appuntamento = new.id_appuntamento; if partecipanti_appuntamento >= capienza_area then signal sqlstate '45000' set message_text = 'Attenzione! Capienza massima dell\'area relativa all\'appuntamento raggiunta'; end if; end// </pre>

19	<p>Un cliente può partecipare solo alle lezioni di attività sportive che rientrano nel suo abbonamento in corso di validità</p>
	<pre> create or replace trigger lezioni_abbonamenti before insert on partecipazioni for each row begin /*Un cliente può partecipare solo alle lezioni di attività sportive che rientrano nel suo abbonamento in corso di validità*/ /*mi dice se l'appuntamento è effettivamente una lezione oppure no*/ declare appuntamento_lezione boolean; /*attività sportiva associata all'appuntamento*/ declare attivita_sportiva_lezione varchar(50); /*id dell'abbonamento in corso di validità per il cliente*/ declare id_abbonamento_corrente integer; /*nome dell'attività sportiva relativa alla lezione se questa è presente tra quelle dell'abbonamento*/ declare attivita_sportiva_abbonamento_corrente varchar(50); /*data dell'appuntamento*/ declare data_appuntamento date; select lezione_bool, nome_attivita_sportiva, data into appuntamento_lezione, attivita_sportiva_lezione, data_appuntamento from appuntamenti where id = new.id_appuntamento; /*l'appuntamento è effettivamente una lezione*/ if appuntamento_lezione then /*sto assumendo che l'abbonamento sia solo uno*/ select id into id_abbonamento_corrente from abbonamenti where cf_cliente = new.cf_cliente AND datediff(data_appuntamento, data_inizio) < durata; if id_abbonamento_corrente is null then signal sqlstate '45000' set message_text = 'Attenzione! Il cliente non ha un abbonamento in corso di validità'; </pre>

	<pre> else select nome_attivita_sportiva into attivita_sportiva_abbonamento_corrente from abbonamenti_attivita_sportive where id_abbonamento = id_abbonamento_corrente AND nome_attivita_sportiva = attivita_sportiva_lezione; /*l'attività sportiva selezionata non rientra tra quelle dell'abbonamento corrente*/ if attivita_sportiva_abbonamento_corrente is null then signal sqlstate '45000' set message_text = 'Attenzione! Il cliente non è iscritto all\'attività sportiva selezionata e non può partecipare a lezioni relative ad essa'; end if; end if; end if; end// </pre>
--	---

20	<p>Uno spazio non può avere come id_categoria_spazio un id relativo a 'sala' o 'campo' , ma solo ai figli oppure ad uno spogliatoio</p>
	<pre> create or replace trigger attivita_sportiva_spazio_associato /*Controllare che le lezioni per attività sportive svolte in sala abbiano come struttura di svolgimento la sala e idem per il campo da gioco. */ before insert on attivita_sportive_spazi for each row begin declare campo bool; declare categoria_spazio_inserito integer; declare id_tipologia_padre_spazio_inserito integer; declare id_sala integer; declare id_campo integer; select id into id_sala from categoria_spazi where tipologia = 'sala' ; select id into id_campo from categoria_spazi </pre>

	<pre> where tipologia = 'campo' ; select categoria_spazio_associato into campo from attivita_sportive where nome = new.nome_attivita_sportiva ; select id_categoria_spazio into categoria_spazio_inserito from spazi where nome = new.nome_spazio; select tipologia_padre into id_tipologia_padre_spazio_inserito from categoria_spazi where id = categoria_spazio_inserito; if campo then if id_tipologia_padre_spazio_inserito = id_sala then signal sqlstate '45000' set message_text ='Attenzione! L\'attività sportiva deve svolgersi in un campo.'; end if; else if id_tipologia_padre_spazio_inserito = id_campo then signal sqlstate '45000' set message_text ='Attenzione! L\'attività sportiva deve svolgersi in una sala.'; end if; end if; end// </pre>
--	---

21	<p>Un appuntamento che è una lezione deve avere obbligatoriamente un istruttore che la supervisiona.</p>
----	---

	<pre> create or replace trigger lezioni_istruttore before insert on appuntamenti for each row begin if new.lezione_bool then if new.cf_istruttore is null then signal sqlstate '45000' set message_text = 'Attenzione! Una lezione deve essere supervisionata da un istruttore'; end if; end if; end// </pre>
--	---

22	<p>Un abbonamento non può avere associata una visita medica con data superiore alla data dell'abbonamento stesso</p>
	<pre> create or replace trigger abbonamento_data_visita_medica_data before insert on abbonamenti for each row begin declare data_visita_medica date; select data into data_visita_medica from visite_mediche where id = new.id_visita_medica; if datediff(data_inizio,data_visita_medica) < 0 then signal sqlstate '45000' set message_text ='Attenzione! Un abbonamento non può avere associata una visita medica con data superiore alla data dell'abbonamento stesso.'; end if; end// </pre>

23	<p>Un appuntamento che è una lezione non deve avere associato un cliente che prenota , se invece è una prenotazione deve avere obbligatoriamente un cliente che prenota</p>
-----------	--

	<pre> create or replace trigger appuntamento_check before insert on appuntamenti for each row BEGIN if new.lezione_bool = 0 AND new.cf_prenotazione IS NULL then signal sqlstate '45000' set message_text = 'Attenzione! Una prenotazione deve avere anche il codice fiscale del cliente che la prenota!'; end if; if new.lezione_bool = 1 AND new.cf_prenotazione IS NOT NULL then signal sqlstate '45000' set message_text = 'Attenzione! Una lezione non può avere un cliente che la prenota'; end if; end// </pre>
--	---

24	<p>Un appuntamento può avere come giorno un valore compreso in questo insieme = {lunedì, martedì, mercoledì, giovedì, venerdì, sabato, domenica}</p>
	<pre> create or replace trigger giorno_della_settimana_appuntamento before insert on appuntamenti for each row begin if new.giorno not in ('lunedì' , 'martedì' , 'mercoledì' , 'giovedì' , 'venerdì' , 'sabato' , 'domenica') then signal sqlstate '45000' set message_text = 'Attenzione! Un appuntamento può avere come giorno un valore compreso in questo insieme = {lunedì, martedì, mercoledì, giovedì, venerdì, sabato, domenica}'; end if; end// </pre>

OPERAZIONI RELATIVE ALL'INTEGRITA' DEI DATI ED ALTRI

11	<p>Il numero di iscritti ad una attività sportiva deve essere minore della capacità massima</p>
-----------	--

	<pre> /*Il numero di iscritti ad una attività sportiva deve essere minore della capacità massima*/ CREATE or replace TRIGGER max_partecipanti_attivita_sportive before insert ON abbonamenti_attivita_sportive for each row BEGIN /*numero di iscritti all'attività sportiva che sto inserendo*/ declare num_iscr integer; declare max_part integer; /*darà sempre una sola riga in quanto il nome dell'attività sportiva è chiave primaria ed io sto inserendo uno di quelli*/ select numero_iscritti , max_partecipanti into num_iscr, max_part from attivita_sportive where nome = new.nome_attivita_sportiva; if num_iscr = max_part then signal sqlstate '45000' set message_text ='Attenzione! L\'attività sportiva selezionata ha raggiunto il numero massimo di partecipanti'; end if; END// </pre>
--	--

16	<p>AGGIORNAMENTO DEL NUMERO DI PARTECIPANTI ALL'INSERIMENTO DI UN'ATTIVITÀ SPORTIVA TRA QUELLE DI UN ABBONAMENO</p>
	<pre> create or replace trigger attivita_sportive_iscritti_handler after insert on abbonamenti_attivita_sportive for each row BEGIN update attivita_sportive set numero_iscritti = numero_iscritti + 1 where new.nome_attivita_sportiva = nome; end// </pre>

25	<p>Aggiornamento del numero di iscritti alle attività sportive allo scadere degli abbonamenti</p>
----	---

Procedure	<pre> Create or replace procedure update_numero_iscritti () BEGIN declare finito boolean default 0; declare attività_sportiva varchar(100); declare count integer; declare cur cursor for select nome_attività_sportiva,count(*) from abbonamenti a inner join abbonamenti_attività_sportive on id = id_abbonamento where DATEDIFF(now(), a.data_inizio) - a.durata = 0 group by nome_attività_sportiva; declare continue handler for sqlstate '02000' set finito = 1; open cur; fetch cur into attività_sportiva,count; ciclo : while finito = 0 do update attività_sportive set numero_iscritti = numero_iscritti - count where nome = attività_sportiva ; fetch cur into attività_sportiva,count; end while ciclo; END// </pre>
Event	<pre> create or replace event abbonamenti_in_scadenza on schedule EVERY 1 DAY do BEGIN call update_numero_iscritti(); END// </pre>

5.2.1 Operazioni richieste

Triggers

OPERAZIONE 11	L'OPERAZIONE NUMERO 11 È IMPLEMENTATA CON UN TRIGGER. DI SEGUITO IL CODICE
DESCRIZIONE	Controllare che le lezioni per attività sportive svolte in sala abbiano come struttura di svolgimento la sala e idem per il campo da gioco .
	<pre>create or replace trigger attivita_sportiva_spazio_associato /*Controllare che le lezioni per attività sportive svolte in sala abbiano come struttura di svolgimento la sala e idem per il campo da gioco. */ before insert on attivita_sportive_spazi for each row begin declare campo bool; declare categoria_spazio_inserito integer; declare id_tipologia_padre_spazio_inserito integer; declare id_sala integer; declare id_campo integer; select id into id_sala from categoria_spazi where tipologia = 'sala' ; select id into id_campo from categoria_spazi where tipologia = 'campo' ; select categoria_spazio_associato into campo from attivita_sportive where nome = new.nome_attivita_sportiva ; select id_categoria_spazio into categoria_spazio_inserito from spazi where nome = new.nome_spazio; select tipologia_padre into id_tipologia_padre_spazio_inserito from categoria_spazi where id = categoria_spazio_inserito; if campo then if id_tipologia_padre_spazio_inserito = id_sala then</pre>

	<pre> signal sqlstate '45000' set message_text ='Attenzione! L\'attività sportiva deve svolgersi in un campo.'; end if; else if id_tipologia_padre_spazio_inserito = id_campo then signal sqlstate '45000' set message_text ='Attenzione! L\'attività sportiva deve svolgersi in una sala.'; end if; end if; end// delimiter ; </pre>
--	--

Stored routine

OPERAZIONE 1	L'OPERAZIONE NUMERO 1 È IMPLEMENTATA CON UNA STORED PROCEDURE. DI SEGUITO IL CODICE
DESCRIZIONE	INSERIMENTO DATI DI UN CLIENTE
	<pre> create or replace procedure Insert_cliente(cf char(16),cognome varchar(30),nome varchar(30),telefono varchar(15) ,email varchar(50) , luogo_nascita varchar(50) , data_nascita date ,sesso boolean) begin insert into clienti values(cf,cognome,nome,telefono,email,luogo_nascita,data_nascita,sesso); end// </pre>

OPERAZIONE 2	L'OPERAZIONE NUMERO 2 È IMPLEMENTATA CON UNA STORED PROCEDURE. DI SEGUITO IL CODICE
-----------------	--

DESCRIZIONE	MODIFICARE L'ORARIO DI UNA LEZIONE.
	<pre> create or replace procedure Update_appuntamento(id_appuntamento integer,orario time) begin update appuntamenti set orario_inizio = orario where id = id_appuntamento ; end// </pre>

OPERAZIONE 3	L'OPERAZIONE NUMERO 3 È IMPLEMENTATA CON UNA STORED PROCEDURE. DI SEGUITO IL CODICE
DESCRIZIONE	ELIMINARE I DATI RELATIVI AD UN DETERMINATO MEDICO.
	<pre> delimiter // create or replace procedure Delete_medico(cf_medico1 char(16),cf_medico2 char(16)) begin declare id_medico integer; declare eliminato1 bool default 0; declare eliminato2 bool default 0; declare finito bool default 0 ; declare id_visita_medica_da_modificare integer; declare cursore cursor for select id from visite_mediche where cf_staff = cf_medico1 and datediff(now(),data) <= 0; declare continue handler for sqlstate '02000' set finito = 1 ; select id into id_medico from ruoli where tipologia = 'medico' ; if cf_medico1 not in(select cf from staff where id_ruolo = id_medico) then signal sqlstate '45000' set message_text = 'Attenzione! Il medico1 non è un medico'; end if; if cf_medico2 not in(select cf from staff where id_ruolo = id_medico) then </pre>

```
        signal sqlstate '45000' set message_text = 'Attenzione! Il medico2 non
è un medico';
    end if;

    select eliminato into eliminato1
    from staff
    where cf = cf_medico1 ;

    select eliminato into eliminato2
    from staff
    where cf = cf_medico2 ;

    if eliminato1 = 1
    then
        signal sqlstate '45000' set message_text = 'Attenzione! Il medico1 è
stato già eliminato';
    end if;

    if eliminato2 = 1
    then
        signal sqlstate '45000' set message_text = 'Attenzione! Il medico2 è
stato eliminato';
    end if;

    update staff
    set eliminato = true
    where cf = cf_medico1 ;

    open cursore;

    fetch cursore into id_visita_medica_da_modificare;
    ciclo : while finito = 0 do
    update visite_mediche
    set cf_staff = cf_medico2
    where id = id_visita_medica_da_modificare ;
    fetch cursore into id_visita_medica_da_modificare;
    end while ciclo ;

    close cursore;
end//
```


OPERAZIONE 12	L'OPERAZIONE NUMERO 12 È IMPLEMENTATA UTILIZZANDO LO SCHEDULER (QUINDI UN EVENTO CON RICORRENZA ANNUALE CHE RICHIAMA UNA STORED ROUTINE). DI SEGUITO IL CODICE
DESCRIZIONE	SNELLIRE LE PRENOTAZIONI SPOSTANDO LE VECCHIE DI OLTRE UN ANNO IN UNA ENTITÀ SEPARATA (PRENOTAZIONI_STORICHE). FARE QUESTA OPERAZIONE OGNI FINE ANNO
PROCEDURE	<pre> create or replace procedure Move_prenotazioni_storiche () begin declare id1 integer; declare giorno1 varchar(20); declare orario_inizio1 time; declare durata1 integer; declare data1 date; declare nome_attivita_sportiva1 varchar(50); declare cf_istruttore1 char(16); declare nome_spazio1 varchar(50); declare finito boolean default 0; declare cursore cursor for select id,giorno,orario_inizio,durata,data,nome_attivita_sportiva,cf_istruttore,nome_spazio from appuntamenti where lezione_bool = 0 and datediff(now(),data) >= 365; declare continue handler for sqlstate '02000' set finito = 1 ; open cursore; fetch cursore into id1,giorno1,orario_inizio1,durata1,data1,nome_attivita_sportiva1,cf_istruttore1,nome_spazio1 ; ciclo : while finito = 0 do insert into prenotazioni_storiche values(id1,giorno1,orario_inizio1,durata1,data1,nome_attivita_sportiva1,cf_istruttore1,nome_spazio1); fetch cursore into id1,giorno1,orario_inizio1,durata1,data1,nome_attivita_sportiva1,cf_istruttore1,nome_spazio1 ; end while ciclo; delete from partecipazioni where id_appuntamento in (select id from appuntamenti where lezione_bool = 0 and datediff(now(),data) >= 365); delete from appuntamenti where lezione_bool = 0 and datediff(now(),data) >= 365; </pre>

	end//
EVENT	create or replace event Prenotazione_storiche_event on schedule EVERY 1 YEAR do BEGIN call Move_prenotazioni_storiche(); END//

Query

OPERAZIONE 4	L'OPERAZIONE NUMERO 4 È IMPLEMENTATA CON UNA QUERY. DI SEGUITO IL CODICE
DESCRIZIONE	VISUALIZZARE TUTTI LE LEZIONI TENUTE DA UN DETERMINATO ISTRUTTORE IN UN DATO GIORNO DELLA SETTIMANA.
	<pre>select * from appuntamenti where lezione_bool and cf_istruttore = 'gm' and giorno = 'lunedì' ;</pre>

OPERAZIONE 5	L'OPERAZIONE NUMERO 5 È IMPLEMENTATA CON UNA QUERY. DI SEGUITO IL CODICE
DESCRIZIONE	VISUALIZZARE IL NUMERO DI ABBONAMENTI SOTTOSCRITTI DA CIASCUN CLIENTE.
	<pre>select cf_cliente , count(*) from abbonamenti group by cf_cliente;</pre>

OPERAZIONE 6	L'OPERAZIONE NUMERO 6 È IMPLEMENTATA CON UNA QUERY. DI SEGUITO IL CODICE
DESCRIZIONE	VISUALIZZARE IL CLIENTE CHE HA SOTTOSCRITTO L'ABBONAMENTO CON IL NUMERO MASSIMO DI CORSI.

	<pre> select * from clienti where cf in (select cf_cliente from abbonamenti inner join abbonamenti_attivita_sportive on id = id_abbonamento group by id having count(*) >= all (select count(*) from abbonamenti_attivita_sportive group by id_abbonamento)); </pre>

OPERAZIONE 7	L'OPERAZIONE NUMERO 7 È IMPLEMENTATA CON UNA QUERY. DI SEGUITO IL CODICE
DESCRIZIONE	VISUALIZZARE IL CLIENTE CHE HA PRENOTATO LO STESSO CAMPO DA GIOCO DUE VOLTE IN UNA SETTIMANA.
	<p>Salvatore Fiorentino:</p> <pre> select * from clienti c where c.cf in (select cf_prenotazione from appuntamenti a where EXISTS (select * from appuntamenti a2 where a.cf_prenotazione = a2.cf_prenotazione and WEEKOFYEAR(a.`data`) = WEEKOFYEAR(a2.`data`))) </pre> <p>Andrea Cuteri:</p> <pre> select * from clienti where cf in(select a1.cf_prenotazione from appuntamenti as a1 inner join appuntamenti as a2 on a1.cf_prenotazione = a2.cf_prenotazione where weekofyear(a1.data) = weekofyear(a2.data)); </pre>

OPERAZIONE 8	L'OPERAZIONE NUMERO 8 È IMPLEMENTATA CON UNA QUERY. DI SEGUITO IL CODICE
DESCRIZIONE	Visualizzare la sala in cui si svolge il minor numero di attività sportive.
	<pre> select * from spazi where nome in (select nome from spazi inner join categoria_spazi on id = id_categoria_spazio where tipologia_padre = (select id from categoria_spazi where tipologia = 'sala') group by nome having (select count(*) from attivita_sportive_spazi where nome = nome_spazio) <= all (select t1.totale from (select nome , count(*) as totale from spazi inner join categoria_spazi on id = id_categoria_spazio where tipologia_padre = (select id from categoria_spazi where tipologia = 'sala') group by nome) as t1)); </pre>

OPERAZIONE 9	L'OPERAZIONE NUMERO 9 È IMPLEMENTATA CON UNA QUERY. DI SEGUITO IL CODICE
DESCRIZIONE	VISUALIZZARE GLI ISTRUTTORI CHE HANNO SOLO IL BREVETTO DI NUOTO.
	<pre> select * from staff where cf in(select cf from brevetti except select cf from brevetti where nome_attivita_sportiva != 'nuoto'); </pre>

OPERAZIONE 10	L'OPERAZIONE NUMERO 10 È IMPLEMENTATA CON UNA QUERY. DI SEGUITO IL CODICE
DESCRIZIONE	VISUALIZZARE I MEDICI CHE SONO ANCHE CLIENTI DEL CENTRO MA NON SEGUONO ATTIVITÀ CHE SI SVOLGONO IN UN DATO ORARIO.
	<p>Salvatore Fiorentino:</p> <pre> versione b: select * from staff s where s.cf in (select c.cf from clienti c inner join staff s2 on c.cf = s2.cf where s2.id_ruolo = (select id from ruoli where tipologia = 'medico') and not EXISTS (SELECT * from appuntamenti a inner join partecipazioni p on a.id = p.id_appuntamento where p.cf_cliente = c.cf and a.orario_inizio = '12:12:12')); </pre> <p>Andrea Cuteri:</p> <pre> select * from staff where cf in(select clienti_medici.cf from(select s.cf from staff s where s.id_ruolo = (select id from ruoli where tipologia = 'medico') intersect select cf from clienti) as clienti_medici where not exists(select * from appuntamenti inner join partecipazioni on id = id_appuntamento where orario_inizio = '12:12:12' and cf_cliente = clienti_medici.cf)); </pre> <p>Giovanni Marasco:</p> <pre> Select * From staff s Where exists(Select * From clienti c Where c.cf = s.cf AND s.id_ruolo = (select id from ruoli where tipologia = 'medico') AND 0 = (Select count(*) </pre>

	<pre>From appuntamenti app inner join partecipazioni part on app.id = part.id_appuntamento Where part.cf_cliente = c.cf AND app.orario_inizio = '12:12:12'));</pre>
--	--