



UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

CENTRO SPORTIVO PARTHENOPE



Studenti:

Viscillo Nicola
Di Palma Emmanuel
Stella Alessandro

Matricola:

0124002557
0124002607
0124002503

Indice

1	PROGETTAZIONE	3
1.1	Analisi dei requisiti	3
1.2	Glossario	4
1.3	Diagramma EE/R	4
1.4	Diagramma relazionale	5
1.5	Elenco delle operazioni di base	5
1.6	Elenco delle operazioni degli utenti	6
1.7	Use case e descrittore di operazione per le operazioni degli utenti	8
1.8	Elenco dei vincoli di integrità statici	10
1.9	Elenco dei vincoli di integrità dinamici	13
1.10	Verifica di normalità dello schema	14
2	IMPLEMENTAZIONE	15
2.1	Creazione utente	15
2.2	Creazione tabelle	16
2.2.1	Socio	16
2.2.2	Armadietto	16
2.2.3	Abbonamento	16
2.2.4	Attività	16
2.2.5	Interessa	17
2.2.6	Corso	17
2.2.7	Impianto	17
2.2.8	Prenotazione	17
2.2.9	Ditta manutenzione	17
2.2.10	Manutenzione	18
2.2.11	Evento	18
2.2.12	Sponsor	18
2.2.13	Sponsorizza	18
2.2.14	Turno	18
2.2.15	Personale	19
2.2.16	Personale istruttore	19
2.2.17	Personale medico	19
2.2.18	Personale amministrativo	19
2.2.19	Personale addetto alle pulizie	19
2.2.20	Personale addetto alle bar	19
2.2.21	Visita medica	20
2.2.22	Magazzino	20
2.2.23	Bar	20
2.2.24	Lavora	20
2.2.25	Assegna	20
2.3	Popolamento	21
2.4	Triggers	54
2.5	Procedure	56

1 PROGETTAZIONE

1.1 Analisi dei requisiti

Per gestire le informazioni relative al centro sportivo Parthenope, è necessario avere un modo per memorizzare i dettagli dei soci. Al momento dell'iscrizione ad ogni socio verrà assegnato un codice socio univoco e verranno registrati nel database i seguenti attributi: nome, cognome, data di nascita, codice fiscale e indirizzo.

Gli abbonamenti forniti ai clienti sono variegati e molto flessibili: mensili, trimestrali, semestrali, annuali e biennali. A ragion di ciò, è essenziale tenere traccia degli abbonamenti dei soci. Ogni abbonamento sarà identificato da un codice abbonamento e conterrà informazioni come la data di inizio e la sua scadenza, ed infine il costo. Inoltre, ad ogni abbonato verrà assegnato un armadietto temporaneo (che sarà indicato nel sistema).

Ogni nuovo socio dovrà anche sottoporsi ad una visita medica, dopo l'iscrizione, che verrà gestita dai medici presenti in struttura. Dopo la visita dovranno essere registrati i risultati di quest'ultima, il codice univoco della visita, la data.

Le preferenze delle attività nel centro sportivo devono essere registrate. Ogni attività avrà un codice identificativo e un nome, e potrebbe richiedere specifico equipaggiamento.

Per quanto riguarda gli impianti presenti nel centro sportivo, è necessario raccogliere informazioni su di essi, come il nome, se sono dotati di copertura (o il contrario) e il tipo di impianto.

Le prenotazioni degli impianti da parte dei soci devono essere gestite, si può prenotare un impianto o un corso. Ogni prenotazione sarà caratterizzata da una data e un orario specifico, un costo associato.

Inoltre, nel centro sportivo verranno anche organizzati dei corsi, ognuno dei quali verrà guidato da un istruttore. Anche qui avremmo dei dati che necessitano di essere registrati: il nome del corso, la data, l'orario, il giorno, il costo e l'istruttore che lo svolgerà.

Ogni impianto dovrà essere sottoposto obbligatoriamente ad una manutenzione periodica; proprio per questo è importante tenere traccia delle operazioni di manutenzione effettuate. Ogni manutenzione verrà caratterizzata da dei dati che verranno raccolti: una data di inizio e di fine, un costo.

Il centro sportivo ha accordi con varie ditte di manutenzione. Di esse si registra il nome e il contratto offerto.

Nel centro vi sono servizi bar, nei quali lavorano addetti esterni. Ogni bar ha il proprio magazzino. Dei magazzini si registra l'indirizzo, il numero di telefono ed il codice del magazzino; del bar si registrano il nome del bar, l'orario di apertura e l'orario di chiusura.

Verranno anche svolte delle operazioni sul personale del centro sportivo, infatti, sarà necessario raccogliere informazioni sui dipendenti. Il personale avrà un codice identificativo, un nome, un cognome, un indirizzo, uno stipendio e la sua categoria di appartenenza (medico, istruttore, barista, etc.). Le informazioni raccolte differiranno anche in base alla categoria di appartenenza, infatti, per quanto riguarda gli istruttori verrà anche memorizzata l'età e lo sport praticato. Per quanto riguarda i medici, è importante che si registri l'anno di abilitazione e la sua specializzazione. Infine, per ogni addetto bar è necessario registrare il proprio ruolo. Inoltre, sarà importante avere informazioni sul turno di lavoro del personale per una corretta pianificazione. E' importante registrare il codice del turno, l'orario di inizio e di fine del turno, ed il tipo.

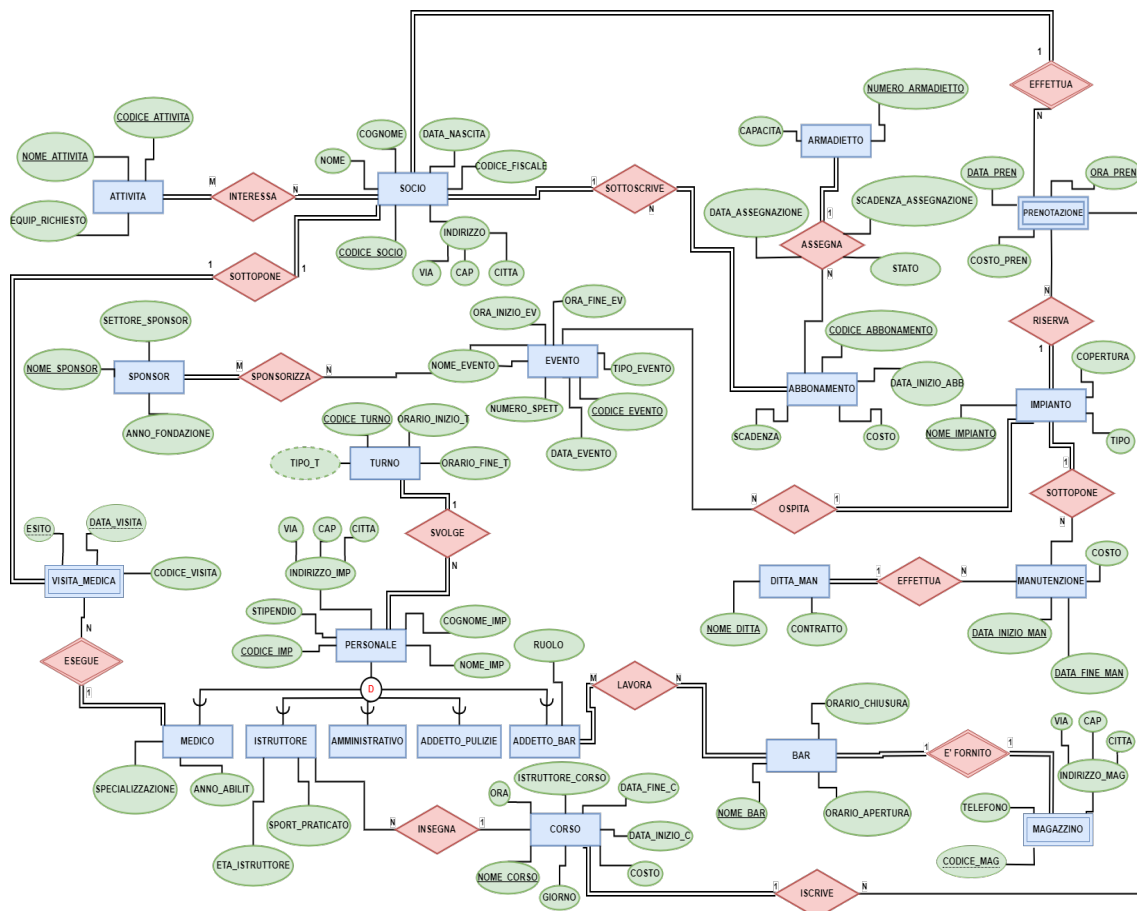
L'impianto sarà anche protagonista di eventi sportivi periodici organizzati da sponsor. Gli eventi sportivi organizzati nel centro sportivo richiedono una registrazione dettagliata. Ogni evento sarà identificato da un codice, un nome, una data, un'ora di inizio e di fine, un tipo di evento e il numero di spettatori previsti.

Infine, anche gli sponsor associati agli eventi devono essere gestiti. Ogni sponsor avrà un nome, un settore di appartenenza e l'anno di fondazione.

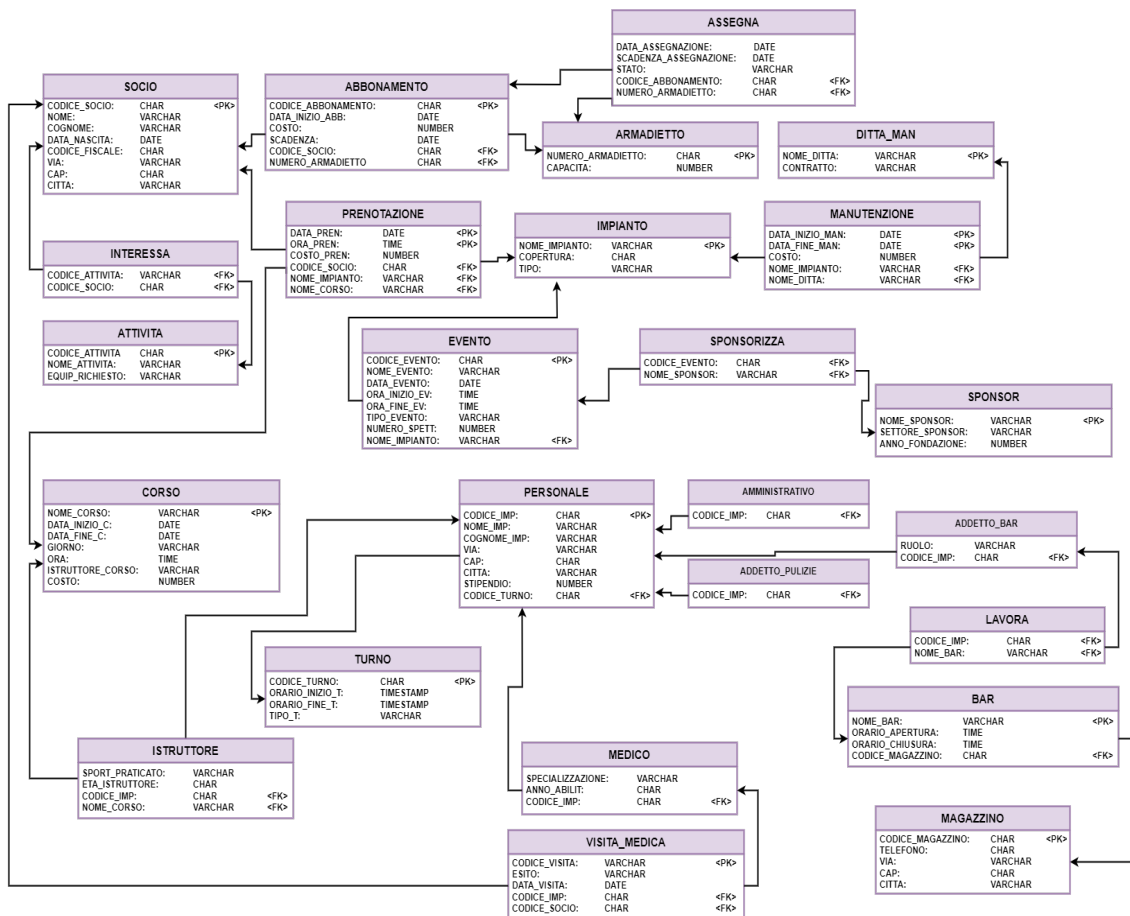
1.2 Glossario

Termine	Definizione	Sinonimo
Abbonamento	Organizzazione e manipolazione dei dati	Rendimento
Armadietto	Struttura per il deposito di oggetti personali	—
Corso	Lezione per insegnare una particolare attività	Lezione, insegnamento
Evento	Attività organizzata	Manifestazione
Impianto	Struttura per una specifica attività	Struttura
Manutenzione	Riparazione, miglioramento del centro	Assistenza tecnica
Personale	Lavoratori del centro	—
Socio	Persona che sottoscrive un abbonamento	Membro, iscritto

1.3 Diagramma EE/R



1.4 Diagramma relazionale



1.5 Elenco delle operazioni di base

1. Inserimento di una tupla:

- Inserimento di un nuovo corso, come la data di inizio, la data di fine, il giorno, l'ora e l'istruttore;
- Inserimento di un nuovo turno, come il codice del turno, orario di inizio, orario di fine e tipo;
- Inserimento di un nuovo evento, come il nome dell'evento, la data, l'ora di inizio, l'ora di fine, il tipo di evento, il numero di spettatori e l'impianto associato;
- Inserimento di una nuova prenotazione, come il nome dell'impianto, la data, l'ora, il costo, il nome del corso e il codice socio;
- Inserimento di una nuova visita medica, come il codice della visita, la data, l'esito, il codice socio e il codice del medico.

2. Aggiornamento di una tupla:

- Aggiornamento delle informazioni di un corso, come la data di inizio, la data di fine, il giorno, l'ora e l'istruttore;
- Aggiornamento delle informazioni di un turno, come il codice del turno, l'orario di inizio, l'orario di fine e il tipo;
- Aggiornamento delle informazioni di un evento, come il nome dell'evento, la data, l'ora di inizio, l'ora di fine, il tipo di evento, il numero di spettatori e l'impianto associato;
- Aggiornamento delle informazioni di una prenotazione, come il nome dell'impianto, la data, l'ora, il costo, il nome del corso e il codice socio;

- Aggiornamento delle informazioni di una visita medica, come il codice della visita, la data, l'esito, il codice socio, il codice del medico.
3. Eliminazione di una tupla:
- Eliminazione delle informazioni di un corso, come la data di inizio, la data di fine, il giorno, l'ora e l'istruttore;
 - Eliminazione delle informazioni di un turno, come il codice del turno, l'orario di inizio, l'orario di fine e il tipo;
 - Eliminazione delle informazioni di un evento, come il nome dell'evento, la data, l'ora di inizio, l'ora di fine, il tipo di evento, il numero di spettatori e l'impianto associato;
 - Eliminazione delle informazioni di una prenotazione, come il nome dell'impianto, la data, l'ora, il costo, il nome del corso e il codice socio;
 - Eliminazione delle informazioni di una visita medica, come il codice della visita, la data, l'esito, il codice socio e il codice del medico.
4. Visualizzazione delle tuple:
- Visualizzazione delle informazioni di un socio, come il nome, il cognome, la data di nascita, il codice fiscale, l'indirizzo, il cap e la città;
 - Visualizzazione delle informazioni di un abbonamento, come la data di inizio, il costo, la scadenza, il codice socio associato e il numero dell'armadietto associato;
 - Visualizzazione delle informazioni di un corso, come il nome del corso, la data di inizio, la data di fine, il giorno, l'ora, l'istruttore e il costo;
 - Visualizzazione delle informazioni di un evento, come il codice evento, il nome dell'evento, la data, l'orario di inizio, l'ora di fine, il tipo di evento, il numero di spettatori e l'impianto associato;
 - Visualizzazione delle informazioni di un armadietto, come il numero dell'armadietto e la capacità;
 - Visualizzazione delle informazioni di un turno, come il codice del turno, orario di inizio, orario di fine e il tipo;
 - Visualizzazione delle informazioni di un impianto, come il nome dell'impianto, la copertura e il tipo;
 - Visualizzazione delle informazioni di una prenotazione, come il nome dell'impianto, la data, l'ora, il costo, il nome del corso e il codice socio;
 - Visualizzazione delle informazioni di una visita medica, come il codice della visita, la data, l'esito, il codice socio e il codice del medico;
 - Visualizzazione delle informazioni di uno sponsor, come il nome dello sponsor, il settore e l'anno di fondazione;
 - Visualizzazione delle informazioni di una ditta di manutenzione, come il nome della ditta e il suo contratto;
 - Visualizzazione delle informazioni di una manutenzione, come la data di inizio e di fine di una manutenzione, il costo della manutenzione, il nome dell'impianto e il nome della ditta manutentrice;
 - Visualizzazione delle informazioni di un bar, come il nome, gli orari di apertura e chiusura e il codice del magazzino associato.

1.6 Elenco delle operazioni degli utenti

Operazioni dell'utente impiegato amministrativo:

- Procedura "aggiungi_corso": permette di aggiungere un nuovo corso e assegnare il suo istruttore;
- Procedura "cancella_prenotazione": permette di cancellare una prenotazione;

- Procedura "aggiorna_prenotazione": permette di aggiornare le informazioni di una prenotazione;
- Procedura "aggiungi_ditta_manutenzione": permette di aggiungere una nuova ditta di manutenzione.

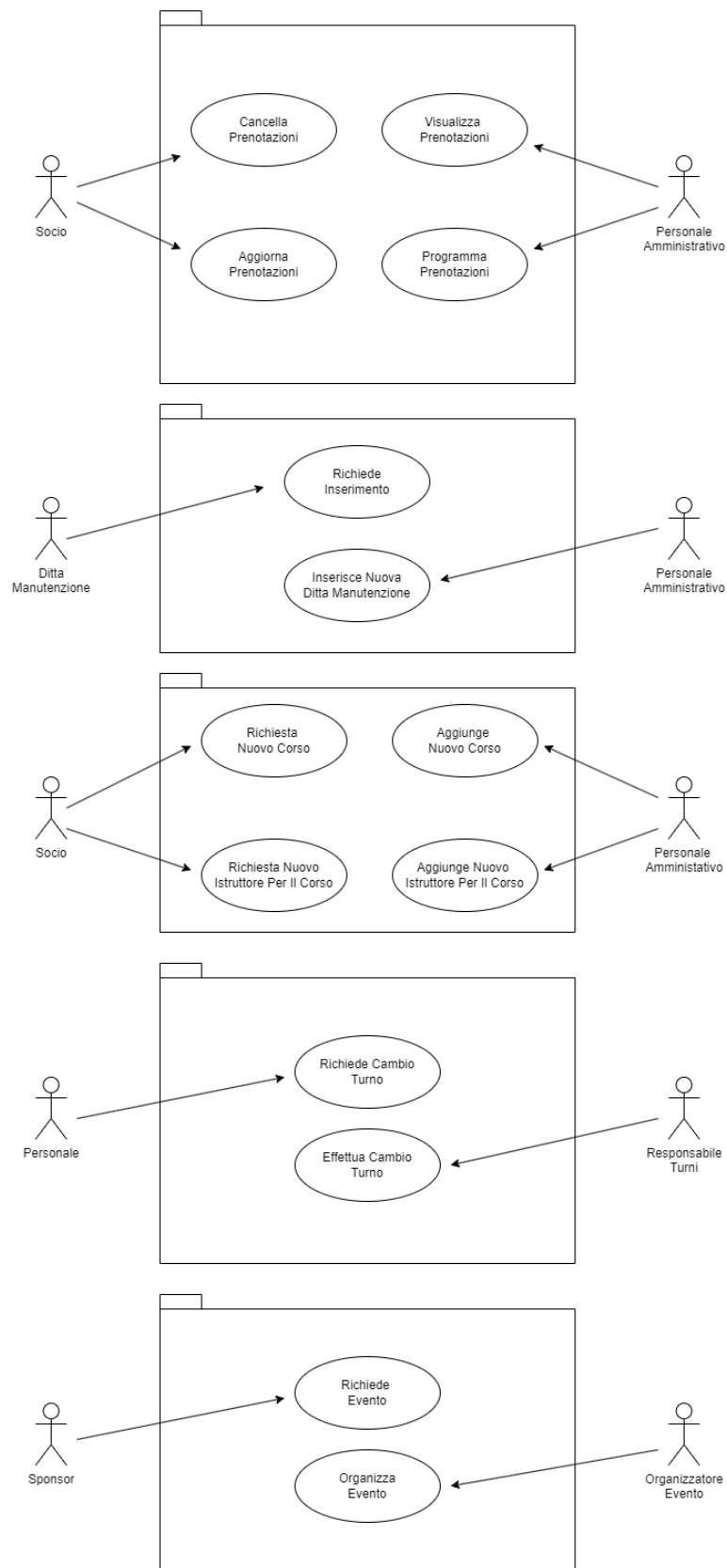
Operazioni dell'utente responsabile dei turni:

- Procedura "AggiornaTurno": permette di aggiornare gli orari di un turno.

Operazioni dell'utente organizzatore di eventi:

- Procedura "CreaEvento": permette di creare un nuovo evento.

1.7 Use case e descrittore di operazione per le operazioni degli utenti



Le schede descrittore delle operazioni degli utenti sono le seguenti.
Le operazioni dell'utente impiegato amministrativo sono:

1. Operazione: aggiungi_corso

- Scopo: permette di aggiungere un nuovo corso e assegnare il suo istruttore;
- Argomenti: nome del corso, descrizione, data di inizio, data di fine, istruttore;
- Risultato: aggiunta del nuovo corso con l'istruttore assegnato o generazione di un errore;
- Errori: istruttore non valido, date in conflitto con altri corsi;
- Dipendenze: corso, istruttore;
- Modifiche: corso.

2. Operazione: cancella_prenotazione

- Scopo: permette di cancellare una prenotazione;
- Argomenti: ID della prenotazione;
- Risultato: cancellazione della prenotazione specificata o generazione di un errore;
- Errori: prenotazione non trovata;
- Dipendenze: prenotazione;
- Modifiche: prenotazione.

3. Operazione: aggiorna_prenotazione

- Scopo: permettere di aggiornare le informazioni di una prenotazione;
- Argomenti: ID della prenotazione, nuove informazioni (data, ora, durata, etc.);
- Risultato: aggiornamento delle informazioni della prenotazione specificata o generazione di un errore;
- Errori: prenotazione non trovata, informazioni non valide;
- Dipendenze: prenotazione;
- Modifiche: prenotazione.

4. Operazione: aggiungi_ditta_manutenzione

- Scopo: permette di aggiungere una nuova ditta di manutenzione;
- Argomenti: nome della ditta, contatti, descrizione dei servizi offerti;
- Risultato: aggiunta della nuova ditta di manutenzione o generazione di un errore;
- Errori: ditta già presente, informazioni mancanti o non valide;
- Dipendenze: ditta_manutenzione;
- Modifiche: ditta_manutenzione.

Le operazioni dell'utente responsabile dei turni sono:

1. Operazione: AggiornaTurno

- Scopo: permette di aggiornare gli orari di un turno;
- Argomenti: ID del turno, nuovi orari di inizio e fine;
- Risultato: aggiornamento degli orari del turno specificato o generazione di un errore;
- Errori: turno non trovato, orari non validi o in conflitto con altri turni;
- Dipendenze: turno;
- Modifiche: turno.

Le operazioni dell'utente organizzatore di eventi sono:

1. Operazione: CreaEvento

- Scopo: permette di creare un nuovo evento;
- Argomenti: nome dell'evento, data, orario, luogo, descrizione, lista degli sponsor;
- Risultato: creazione del nuovo evento con le informazioni specificate o generazione di un errore;
- Errori: informazioni incomplete o non valide, sponsor mancanti o non validi;
- Dipendenze: evento, sponsor;
- Modifiche: evento.

1.8 Elenco dei vincoli di integrità statici

Tabella "socio":

- codice_socio è la chiave primaria (**primary key**);
- nome non può essere nullo (**not null**);
- cognome non può essere nullo (**not null**);
- codice_fiscale deve essere unico (**unique**).

Tabella "armadietto":

- codice_abbonamento è la chiave primaria (**primary key**).

Tabella "abbonamento":

- codice_abbonamento è la chiave primaria (**primary key**);
- data_inizio_abb non può essere nullo (**not null**);
- codice_socio_abb non può essere nullo (**not null**);
- numero_armadietto_abb non può essere nullo (**not null**);
- vincoli di chiave esterna (**foreign key**) verso la tabella "socio" (**constraint fk_abbonamento1**) e la tabella "armadietto" (**constraint fk_abbonamento2**) rispettivamente.

Tabella "attivit ":

- codice_attivita_int   la chiave primaria (**primary key**);
- nome_attivita non pu  essere nullo (**not null**).

Tabella "interessa":

- codice_attivita_int e codice_socio_int formano la chiave primaria (**primary key**);
- vincoli di chiave esterna (**foreign key**) verso la tabella "attivit " (**constraint fk_interessa1**) e la tabella "socio" (**constraint fk_interessa2**) rispettivamente.

Tabella "corso":

- nome_corso   la chiave primaria (**primary key**);
- data_inizio_c non pu  essere nullo (**not null**);
- data_fine_c non pu  essere nullo (**not null**);
- giorno non pu  essere nullo (**not null**);
- ora non pu  essere nullo (**not null**).

Tabella "impianto":

- nome_impianto è la chiave primaria (primary key);
- copertura può assumere solo i valori 'aperto', 'chiuso', 'Aperto', 'Chiuso';
- tipo non può essere nullo (not null).

Tabella "prenotazione":

- data_pren e ora_pren formano la chiave primaria (primary key);
- data_pren non può essere nullo (not null);
- ora_pren non può essere nullo (not null);
- costo_pren non può essere nullo (not null);
- codice_socio_pren non può essere nullo (not null);
- vincoli di chiave esterna (foreign key) verso la tabella "socio" (constraint fk_prenotazione1), la tabella "impianto" (constraint fk_prenotazione2) e la tabella "corso" (constraint fk_prenotazione3 rispettivamente).

Tabella "ditta_man":

- nome_ditta è la chiave primaria (primary key).

Tabella "manutenzione":

- data_inizio_man e data_fine_man formano la chiave primaria (primary key);
- data_inizio_man non può essere nullo (not null);
- data_fine_man non può essere nullo (not null);
- costo non può essere nullo (not null);
- nome_impianto_man non può essere nullo (not null);
- nome_ditta_man non può essere nullo (not null);
- vincoli di chiave esterna (foreign key) verso la tabella "impianto" (constraint fk_manutenzione1) e la tabella "ditta_man" (constraint fk_manutenzione2) rispettivamente.

Tabella "evento":

- codice_evento è la chiave primaria (primary key);
- data_evento non può essere nullo (not null);
- ora_inizio_ev non può essere nullo (not null);
- ora_fine_ev non può essere nullo (not null);
- nome_impianto_ev non può essere nullo (not null);
- vincolo di chiave esterna (foreign key) verso la tabella "impianto" (constraint fk_evento).

Tabella "sponsor":

- nome_sponsor è la chiave primaria (primary key).

Tabella "sponsorizza":

- codice_evento_sp e nome_sponsor_sp formano la chiave primaria (primary key);
- vincoli di chiave esterna (foreign key) verso la tabella "evento" (constraint fk_sponsorizza1) e la tabella "sponsor" (constraint fk_sponsorizza2) rispettivamente.

Tabella "turno":

- codice_turno è la chiave primaria (primary key);
- orario_inizio_t non può essere nullo (not null);
- orario_fine_t non può essere nullo (not null);
- tipo_t può assumere solo i valori 'Mattina', 'Pomeriggio', 'Sera', 'mattina', 'pomeriggio', 'sera'.

Tabella "personale":

- codice_imp è la chiave primaria (primary key);
- nome_imp non può essere nullo (not null);
- cognome_imp non può essere nullo (not null);
- codice_turno_p non può essere nullo (not null);
- vincolo di chiave esterna (foreign key) verso la tabella "turno" (constraint fk_personale).

Tabella "istruttore":

- codice_imp_istr è la chiave primaria (primary key);
- nome_corso_istr non può essere nullo (not null);
- vincoli di chiave esterna (foreign key) verso la tabella "personale" (constraint fk_istruttore1) e la tabella "corso" (constraint fk_istruttore2) rispettivamente.

Tabella "medico":

- codice_imp_med è la chiave primaria (primary key);
- specializzazione non può essere nullo (not null);
- vincolo di chiave esterna (foreign key) verso la tabella "personale" (constraint fk_medico).

Tabella "amministrativo":

- codice_imp_amm è la chiave primaria (primary key);
- vincolo di chiave esterna (foreign key) verso la tabella "personale" (constraint fk_amministrativo).

Tabella "addetto_pulizie":

- codice_imp_add è la chiave primaria (primary key);
- vincolo di chiave esterna (foreign key) verso la tabella "personale" (constraint fk_addetto).

Tabella "addetto_bar":

- codice_imp_addbar è la chiave primaria (primary key);
- ruolo non può essere nullo (not null);
- vincolo di chiave esterna (foreign key) verso la tabella "personale" (constraint fk_addetto_bar).

Tabella "visita_medica":

- codice_visita è la chiave primaria (**primary key**);
- esito non può essere nullo (**not null**);
- data_visita non può essere nullo (**not null**);
- codice_imp_med_vis non può essere nullo (**not null**);
- codice_socio_vis non può essere nullo (**not null**);
- vincoli di chiave esterna (**foreign key**) verso la tabella "medico" (**constraint fk_visita1**) e la tabella "socio" (**constraint fk_visita2**) rispettivamente.

Tabella "magazzino":

- codice_magazzino è la chiave primaria (**primary key**);
- telefono è unico (**unique**).

Tabella "bar":

- nome_bar è la chiave primaria (**primary key**);
- orario_apertura non può essere nullo (**not null**);
- orario_chiusura non può essere nullo (**not null**);
- codice_magazzino_bar non può essere nullo (**not null**);
- vincolo di chiave esterna (**foreign key**) verso la tabella "magazzino" (**constraint fk_bar2**).

Tabella "lavora":

- codice_imp_addbar_lavora e nome_bar_lavora formano la chiave primaria (**primary key**);
- vincoli di chiave esterna (**foreign key**) verso la tabella "addetto_bar" (**constraint fk_lavora1**) e la tabella "bar" (**constraint fk_lavora2**) rispettivamente.

1.9 Elenco dei vincoli di integrità dinamici

- Il socio non può avere più di un abbonamento attivo alla volta;
- Il numero dei soci non può superare il numero di armadietti disponibili;
- Il campo "tipo" nella tabella "impianto" può contenere solo i valori selezionati;
- Il campo "contratto" nella tabella "ditta_man" può contenere solo i valori selezionati;
- Il campo "sport_praticato" nella tabella "istruttore" può contenere solo i valori selezionati;
- La data di inizio di un corso deve essere precedente della data di fine;
- Il socio non può effettuare più di una visita medica al giorno;
- La data di inizio della manutenzione deve essere precedente della data di fine;
- La data di inizio di un abbonamento deve essere precedente della scadenza.

1.10 Verifica di normalità dello schema

La normalizzazione è un processo di progettazione del database che mira ad organizzare le tabelle in modo efficiente e senza ridondanze, riducendo al minimo l'inserimento, l'aggiornamento e la cancellazione di dati anomali. Esistono diversi livelli di normalizzazione, ognuno dei quali stabilisce determinate regole per la struttura delle tabelle:

1. **Prima forma normale (1NF):** la Prima Forma Normale richiede che ogni tabella abbia una chiave primaria unica (PK) che identifica in modo univoco ogni riga della tabella. Inoltre, ogni attributo all'interno della tabella deve contenere solo valori atomici, cioè non può essere suddiviso ulteriormente. Infine, ogni attributo deve avere un nome univoco all'interno della tabella. Tutti gli attributi sembrano essere atomici e non ci sono dipendenze funzionali multiattributo o dipendenze funzionali parziali;
2. **Seconda forma normale (2NF):** la Seconda Forma Normale richiede che una tabella soddisfi la 1NF e che tutti gli attributi non chiave dipendano completamente dalla chiave primaria. Ciò significa che ogni attributo non chiave deve dipendere dalla chiave primaria in modo diretto, non transitivo. Se una tabella ha una chiave primaria composta da più attributi, ogni attributo non chiave deve dipendere da tutta la chiave primaria. Per verificare la 2NF, è necessario identificare le dipendenze funzionali tra gli attributi delle tabelle. Le uniche chiavi multiattributo sono quelle di prenotazione e manutenzione, che non determinano in maniera parziale nessun attributo della relazione. Le tabelle soddisfano la seconda forma normale, poiché ogni attributo non chiave dipende completamente dalla chiave primaria e non ci sono dipendenze funzionali tra attributi non chiave;
3. **Terza forma normale (3NF):** la Terza Forma Normale richiede che una tabella soddisfi la 2NF e che non ci siano dipendenze funzionali tra gli attributi non chiave. Ciò significa che gli attributi non chiave devono dipendere solo dalla chiave primaria o da altre chiavi alternative, ma non da altri attributi non chiave. Per eseguire la verifica della 3NF, è necessario esaminare le dipendenze funzionali tra gli attributi delle tabelle. Le tabelle soddisfano la terza forma normale, poiché ogni attributo non chiave dipende completamente dalla chiave primaria e non ci sono dipendenze funzionali tra attributi non chiave;
4. **Forma normale di Boyce-Codd (BCNF):** la Forma Normale di Boyce-Codd (BCNF) richiede che una tabella soddisfi la 3NF e che ogni dipendenza funzionale non banale implichi che la chiave primaria sia una superchiave della tabella. In altre parole, se esiste una dipendenza funzionale non banale $X \rightarrow Y$, dove Y non è una sottochiave di X , allora X deve essere una superchiave. Tutte le tabelle soddisfano la BCNF, poiché l'attributo chiave di ciascuna tabella determina tutti gli altri attributi presenti nella tabella.

2 IMPLEMENTAZIONE

2.1 Creazione utente

– CREAZIONE DEGLI UTENTI

```
create user db_parthenope identified by admin;
create user socio identified by SocioPass123;
create user istruttore identified by IstPass123;
create user addetto_bar identified by AddbPass123;
create user medico identified by MedPass123;
create user addetto_pulizie identified by AddPPass123;
create user impiegato_amministrativo identified by AmmPass123;
create user ditta_manutenzione identified by DittaManPass123;
create user responsabile_turni identified by RespTurPass123;
create user sponsors identified by SponsorsPass123;
create user organizzatore_evento identified by OrgEvPass123;
```

– ASSEGNAZIONE DEI PRIVILEGI AGLI UTENTI

```
grant all privileges to db_parthenope;
grant connect, resource, dba to db_parthenope;

grant create session to socio;
grant select on armadietto to socio;
grant select on abbonamento to socio;
grant select on prenotazione to socio;

grant create session to istruttore;
grant select, insert, update, delete on corso to istruttore;
grant select on turno to istruttore;

grant create session to addetto_bar;
grant select on bar to addetto_bar;
grant select on magazzino to addetto_bar;
grant select on turno to addetto_bar;

grant create session to medico;
grant select, insert, update, delete on visita_medica to medico;
grant select on turno to medico;

grant create session to addetto_pulizie;
grant select on bar to addetto_pulizie;
grant select on impianto to addetto_pulizie;
grant select on turno to addetto_pulizie;

grant create session to impiegato_amministrativo;
grant select, insert, update, delete on prenotazione to impiegato_amministrativo;
grant select, insert, update, delete on turno to impiegato_amministrativo;
grant select, insert, update, delete on corso to impiegato_amministrativo;
grant execute on aggiungi_ditta_manutenzione to impiegato_amministrativo;
grant execute on aggiungi_corso to impiegato_amministrativo;
grant execute on aggiorna_prenotazione to impiegato_amministrativo;
grant execute on cancella_prenotazione to impiegato_amministrativo;

grant create session to ditta_manutenzione;
grant create session to ditta_man to ditta_manutenzione;
grant select on manutenzione to ditta_manutenzione;

grant create session to responsabile_turni;
grant select, insert, update, delete on turno to responsabile_turni;
grant execute on AggiornaTurno to responsabile_turni;

grant create session to organizzatore_evento;
grant select, insert, update, delete on evento to organizzatore_evento;
grant select on impianto to organizzatore_evento;
```

```

grant select on sponsor to organizzatore;
grant execute on CreaEvento to organizzatore_evento;

grant create session to sponsors;
grant select on sponsor to sponsors;
grant select on impianto to sponsors;
grant select on sponsor to sponsors;

```

2.2 Creazione tabelle

2.2.1 Socio

```

create table socio (
    codice_socio      char(9)          primary key,
    nome              varchar(30)       not null,
    cognome           varchar(30)       not null,
    data_nascita       date,
    codice_fiscale     char(16)         unique,
    via               varchar(30),
    cap               char(5),
    citta             varchar(30)
)

```

2.2.2 Armadietto

```

create table armadietto (
    numero_armadietto char(2)          primary key,
    capacita          number
)

```

2.2.3 Abbonamento

```

create table abbonamento (
    codice_abbonamento char(9)          primary key,
    data_inizio_abb      date            not null,
    costo                number,
    scadenza             date            not null,
    codice_socio_abb     char(9)          not null,
    numero_armadietto_abb char(2)         not null,
    constraint fk_abbonamento1 foreign key (codice_socio_abb) references
    socio(codice_socio),
    constraint fk_abbonamento2 foreign key (numero_armadietto_abb) references
    armadietto(numero_armadietto)
)

```

2.2.4 Attività

```

create table attivita (
    codice_attivita     char(9)          primary key,
    nome_attivita       varchar(30)       not null,
    equip_richiesto     varchar(30)
)

```


2.2.5 Interessea

```
create table interessa (
    codice_attivita_int      char(9)          not null,
    codice_socio_int         char(9)          not null,
    constraint pk_interessa primary key (codice_attivita_int, codice_socio_int),
    constraint fk_interessa1 foreign key (codice_attivita_int) references
    attivita(codice_attivita),
    constraint fk_interessa2 foreign key (codice_socio_int) references
    socio(codice_socio)
)
```

2.2.6 Corso

```
create table corso (
    nome_corso              varchar(30)        primary key,
    data_inizio_c           date               not null,
    data_fine_c             date               not null,
    giorno                  varchar(10)        not null,
    ora                     timestamp(0)       not null,
    istruttore_corso       varchar(30),
    costo                   number
)
```

2.2.7 Impianto

```
create table impianto (
    nome_impianto          varchar(30)        primary key,
    copertura              char(6)            check (copertura in ('aperto',
    'chiuso', 'Aperto', 'Chiuso')),
    tipo                   varchar(30)        not null
)
```

2.2.8 Prenotazione

```
create table prenotazione (
    data_pren              date               not null,
    ora_pren               timestamp(0)       not null,
    costo_pren             number             not null,
    codice_socio_pren      char(9)            not null,
    nome_impianto_pren     varchar(30),
    nome_corso_pren        varchar(30),
    constraint pk_prenotazione primary key (data_pren, ora_pren),
    constraint fk_prenotazione1 foreign key (codice_socio_pren) references
    socio(codice_socio),
    constraint fk_prenotazione2 foreign key (nome_impianto_pren) references
    impianto(nome_impianto),
    constraint fk_prenotazione3 foreign key (nome_corso_pren) references
    corso(nome_corso)
)
```

2.2.9 Ditta manutenzione

```
create table ditta_man (
    nome_ditta             varchar(30)        primary key,
    contratto              varchar(30)
)
```

2.2.10 Manutenzione

```
create table manutenzione (  
    data_inizio_man      date          not null,  
    data_fine_man        date          not null,  
    costo                number        not null,  
    nome_impianto_man    varchar(30)   not null,  
    nome_ditta_man       varchar(30)   not null,  
    constraint pk_manutenzione primary key (data_inizio_man, data_fine_man),  
    constraint fk_manutenzione1 foreign key (nome_impianto_man) references  
        impianto(nome_impianto)  
    constraint fk_manutenzione2 foreign key (nome_ditta_man) references  
        ditta_man(nome_ditta)  
)
```

2.2.11 Evento

```
create table evento (  
    codice_evento        char(3)       primary key,  
    nome_evento          varchar(30)   not null,  
    data_evento          date          not null,  
    ora_inizio_ev        timestamp(0)  not null,  
    ora_fine_ev          timestamp(0)  not null,  
    tipo_evento          varchar(30),  
    numero_spett         number,  
    nome_impianto_ev     varchar(30)   not null,  
    constraint fk_evento foreign key (nome_impianto_ev) references  
        impianto(nome_impianto)  
)
```

2.2.12 Sponsor

```
create table sponsor (  
    nome_sponsor         varchar(30)   primary key,  
    settore_sponsor     varchar(30),  
    anno_fondazione      number  
)
```

2.2.13 Sponsorizza

```
create table sponsorizza (  
    codice_evento_sp     char(3)       not null,  
    nome_sponsor_sp      varchar(30)   not null,  
    constraint pk_sponsorizza primary key (codice_evento_sp, nome_evento_sp),  
    constraint fk_sponsorizza1 foreign key (codice_evento_sp) references  
        evento(codice_evento),  
    constraint fk_sponsorizza2 foreign key (nome_sponsor_sp) references  
        sponsor(nome_sponsor)  
)
```

2.2.14 Turno

```
create table turno (  
    codice_turno         char(6)       primary key,  
    orario_inizio_t      timestamp(0)  not null,  
    orario_fine_t        timestamp(0)  not null,  
    tipo_t               varchar(30)   check (tipo_t in ('Mattina',  
        'Pomeriggio', 'Sera', 'mattina', 'pomeriggio', 'sera'))
```

2.2.15 Personale

```
create table personale (  
    codice_imp        char(6)           primary key,  
    nome_imp          varchar(30)       not null,  
    cognome_imp       varchar(30)       not null,  
    via_imp           varchar(30),  
    cap_imp           char(5),  
    citta_imp         varchar(30),  
    stipendio         number,  
    codice_turno_p    char(6)           not null  
)
```

2.2.16 Personale istruttore

```
create table istruttore (  
    codice_imp_istr    char(6)           primary key,  
    sport_praticato    varchar(30),  
    eta_istruttore     char(2),  
    nome_corso_istr    varchar(30)       not null,  
    constraint fk_istruttore1 foreign key (codice_imp_istr) references  
    personale(codice_imp),  
    constraint fk_istruttore2 foreign key (nome_corso_istr) references  
    corso(nome_corso)  
)
```

2.2.17 Personale medico

```
create table medico (  
    codice_imp_med     char(6)           primary key,  
    specializzazione    varchar(30)       not null,  
    anno_abilit        char(4),  
    constraint fk_medico foreign key (codice_imp_med) references  
    personale(codice_imp)  
)
```

2.2.18 Personale amministrativo

```
create table amministrativo (  
    codice_imp_amm      char(6)           primary key,  
    constraint fk_amministrativo foreign key (codice_imp_amm) references  
    personale(codice_imp)  
)
```

2.2.19 Personale addetto alle pulizie

```
create table addetto_pulizie (  
    codice_imp_add      char(6)           primary key,  
    constraint fk_addetto foreign key (codice_imp_add) references  
    personale(codice_imp)  
)
```

2.2.20 Personale addetto alle bar

```
create table addetto_bar (  
    codice_imp_addbar    char(6)           primary key,  
    ruolo               varchar(30)       not null,  
    constraint fk_addetto_bar foreign key (codice_imp_addbar) references  
    personale(codice_imp)  
)
```

2.2.21 Visita medica

```
create table visita_medica (  
    codice_visita        varchar(4)        primary key,  
    esito                varchar(30)        not null,  
    data_visita          date              not null,  
    codice_imp_med_vis    char(6)           not null,  
    codice_socio_vis     char(9)           not null,  
    constraint fk_visita1 foreign key (codice_imp_med_vis) references  
        medico(codice_imp_med)  
    constraint fk_visita2 foreign key (codice_socio_vis) references  
        socio(codice_socio)  
)
```

2.2.22 Magazzino

```
create table magazzino (  
    codice_magazzino     char(1)           primary key,  
    telefono             char(10)          unique,  
    via_m               varchar(30),  
    cap_m               char(5),  
    citta_m             varchar(30)  
)
```

2.2.23 Bar

```
create table bar (  
    nome_bar            varchar(30)        primary key,  
    orario_apertura     timestamp(0)       not null,  
    orario_chiusura     timestamp(0)       not null,  
    codice_magazzino_bar char(1)           not null,  
    constraint fk_bar foreign key (codice_magazzino_bar) references  
        magazzino(codice_magazzino)  
)
```

2.2.24 Lavora

```
create table lavora (  
    codice_imp_addbar_lavora char(6),  
    nome_bar_lavora        varchar(30),  
    constraint pk_lavora primary key (codice_imp_addbar_lavora, nome_bar_lavora),  
    constraint fk_lavora1 foreign key (codice_imp_addbar_lavora) references  
        addetto_bar(codice_imp_addbar),  
    constraint fk_lavora2 foreign key (nome_bar_lavora) references bar(nome_bar)  
)
```

2.2.25 Assegna

```
create table assegna (  
    data_assegnazione    date,  
    scadenza_assegnazione date,  
    stato                varchar(30)       check (stato in ('disponibile',  
        'non disponibile', 'Disponibile', 'Non disponibile')),  
    numero_armadietto_assegna char(2),  
    codice_abbonamento_assegna char(9),  
    constraint pk_assegna primary key (data_assegnazione, scadenza_assegnazione),  
    constraint fk_assegna1 foreign key (numero_armadietto_assegna) references  
        armadietto(numero_armadietto),  
    constraint fk_assegna2 foreign key (codice_abbonamento_assegna) references  
        abbonamento(codice_abbonamento)
```


POPOLAMENTO DELLA TABELLA SOCIO:

```
insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000001', 'Mario', 'Rossi', date '1990-05-15',
'RSSMRI90A15H234B', 'Via Roma 1', '80143', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000002', 'Giulia', 'Bianchi', date '1985-09-20',
'BNCGLI85P20H456C', 'Via Milano 2', '80145', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000004', 'Chiara', 'Mancini', date '1988-03-25',
'MNCCHR88C25H987E', 'Via Napoli 4', '80146', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000005', 'Marco', 'Serra', date '1995-01-05',
'SRRMRC95A05H654F', 'Via Firenze 5', '80147', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000006', 'Laura', 'Leone', date '1991-11-12',
'LNLALR91S12H321G', 'Via Bologna 6', '80148', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000008', 'Francesca', 'Galli', date '1993-06-30',
'GLLFRN93H30H876I', 'Via Genova 8', '80143', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000009', 'Matteo', 'Riva', date '1986-04-18',
'RVAMTT86D18H234J', 'Via Palermo 9', '80145', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000010', 'Giorgia', 'Marino', date '1994-02-10',
'MRNGRG94B10H567K', 'Via Catania 10', '80144', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000007', 'Simone', 'Bianchi', date '1987-08-08',
'BNCSSM87M08H543H', 'Via Torino 7', '80140', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000011', 'Alessandro', 'Santoro', date '1992-10-22',
'SNTRLSS92R22H890', 'Via Salerno 11', '80146', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000012', 'Sara', 'Gatti', date '1988-07-14',
'GTTSTRA88L14H345M', 'Via Verona 12', '80147', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000013', 'Fabio', 'Lombardi', date '1984-05-28',
'LMBFB084P28H678N', 'Via Padova 13', '80148', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000014', 'Eleonora', 'Fabbri', date '1997-03-03',
'FBBELN97C03H9010', 'Via Pisa 14', '80143', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000015', 'Giovanni', 'Rizzi', date '1990-01-17',
'RZZGVN90A17H234P', 'Via Roma 15', '80145', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000016', 'Stefania', 'Rossi', date '1986-11-11',
'RSSSTF86S11H567Q', 'Via Firenze 16', '80144', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000017', 'Carlo', 'Serra', date '1993-09-07',
'SRRCRL93P07H890R', 'Via Bologna 17', '80149', 'Napoli');
```

```

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000018', 'Roberta', 'Rinaldi', date '1988-06-23',
'RNLRBT88H23H123S', 'Via Napoli 18', '80147', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000019', 'Lorenzo', 'Marini', date '1995-04-12',
'MRNLZN95D12H456T', 'Via Milano 19', '80148', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000020', 'Martina', 'Pellegrini', date '1991-02-04',
'PLGMRT91B04H789U', 'Via Roma 20', '80143', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000021', 'Alessia', 'Ferrari', date '1987-12-16',
'FRRALS87T16H987V', 'Via Venezia 21', '80145', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000022', 'Giacomo', 'Serra', date '1994-10-29',
'SRRGCM94R29H321W', 'Via Napoli 22', '80144', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000023', 'Claudia', 'Rizzo', date '1989-08-25',
'RZZCLD89P25H654X', 'Via Firenze 23', '80146', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000024', 'Federico', 'Rinaldi', date '1996-06-07',
'RNLDFT96H07H876Y', 'Via Bologna 24', '80147', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000025', 'Alessandra', 'Fabbri', date '1992-04-30',
'FBBALS92E30H234Z', 'Via Milano 25', '80148', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000026', 'Matteo', 'Cattaneo', date '1987-03-17',
'CTTNMT87C17H567A', 'Via Genova 26', '80143', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000027', 'Silvia', 'Sartori', date '1993-01-09',
'SRTSLV93M09H890B', 'Via Torino 27', '80145', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000028', 'Ludovica', 'Longo', date '1988-11-21',
'LNGLDV88S21H345C', 'Via Verona 28', '80144', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000029', 'Andrea', 'Palmieri', date '1984-09-04',
'PLMADR84P04H678D', 'Via Padova 29', '80146', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000030', 'Camilla', 'Villa', date '1997-07-26',
'VLLCML97H26H901E', 'Via Pisa 30', '80147', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000031', 'Roberto', 'Rossi', date '1990-05-15',
'RSSRBT90A15H234F', 'Via Roma 31', '80148', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000032', 'Marta', 'Bianchi', date '1985-09-20',
'BNCMRT85P20H456G', 'Via Milano 32', '80143', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000033', 'Simone', 'Russo', date '1992-07-10',
'RSSSMN92L10H789H', 'Via Venezia 33', '80145', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000034', 'Elisa', 'Mancini', date '1988-03-25',
'MNCESA88C25H987I', 'Via Napoli 34', '80144', 'Napoli');

```



```

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000035', 'Davide', 'Serra', date '1995-01-05',
'SRRDVD95A05H654J', 'Via Firenze 35', '80146', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000036', 'Giorgia', 'Leone', date '1991-11-12',
'LNLGIA91S12H321K', 'Via Bologna 36', '80147', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000037', 'Riccardo', 'Bianchi', date '1987-08-08',
'BNCRC87M08H543L', 'Via Torino 37', '80148', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000038', 'Francesca', 'Galli', date '1993-06-30',
'GLLFRN93H30H876M', 'Via Genova 38', '80143', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000039', 'Matteo', 'Riva', date '1986-04-18',
'RVMMTT86D18H234N', 'Via Palermo 39', '80145', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000040', 'Giulia', 'Marino', date '1994-02-10',
'MRNGLI94B10H5670', 'Via Catania 40', '80144', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000042', 'Sara', 'Gatti', date '1988-07-14',
'GTTSTRA88L14H345Q', 'Via Verona 42', '80147', 'Napoli');

insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000043', 'Fabio', 'Lombardi', date '1984-05-28',
'LMBFB084P28H678R', 'Via Padova 43', '80148', 'Napoli');
insert into socio(codice_socio, nome, cognome, data_nascita, codice_fiscale, via,
cap, citta) values ('SOC000044', 'Sabino', 'Malese', date '1985-05-15',
'MLSSBN85E15F839L', 'Via Roma 44', '80144', 'Napoli');

```

POPOLAMENTO DELLA TABELLA ABBONAMENTO:

```

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000001', date '2023-01-01',
50, date '2023-01-31', 'SOC000001', '01');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000002', date '2023-01-01',
120, date '2023-03-31', 'SOC000002', '02');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000003', date '2023-01-01',
250, date '2023-06-30', 'SOC000004', '03');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000004', date '2023-01-01',
450, date '2023-12-31', 'SOC000005', '04');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000005', date '2023-01-01',
750, date '2024-12-31', 'SOC000006', '05');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000006', date '2023-02-01',
50, date '2023-02-28', 'SOC000008', '06');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000007', date '2023-02-01',
120, date '2023-04-30', 'SOC000009', '07');

```



```

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000008', date '2023-02-01',
250, date '2023-07-31', 'SOC000010', '08');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000009', date '2023-02-01',
450, date '2023-12-31', 'SOC000007', '09');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000010', date '2023-02-01',
750, date '2025-01-31', 'SOC000011', '10');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000011', date '2023-03-01',
50, date '2023-03-31', 'SOC000012', '11');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000012', date '2023-03-01',
120, date '2023-05-31', 'SOC000013', '12');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000013', date '2023-03-01',
250, date '2023-08-31', 'SOC000014', '13');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000014', date '2023-03-01',
450, date '2023-12-31', 'SOC000015', '14');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000016', date '2023-04-01',
50, date '2023-04-30', 'SOC000017', '16');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000017', date '2023-04-01',
120, date '2023-06-30', 'SOC000018', '17');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000018', date '2023-04-01',
250, date '2023-09-30', 'SOC000019', '18');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000019', date '2023-04-01',
450, date '2023-12-31', 'SOC000020', '19');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000020', date '2023-04-01',
750, date '2025-03-31', 'SOC000021', '20');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000021', date '2023-05-01',
50, date '2023-05-31', 'SOC000022', '21');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000022', date '2023-05-01',
120, date '2023-07-31', 'SOC000023', '22');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000023', date '2023-05-01',
250, date '2023-10-31', 'SOC000024', '23');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000024', date '2023-05-01',
450, date '2023-12-31', 'SOC000025', '24');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000025', date '2023-05-01',
750, date '2025-04-30', 'SOC000026', '25');

```

```

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000026', date '2023-06-01',
50, date '2023-06-30', 'SOC000027', '26');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000027', date '2023-06-01',
120, date '2023-08-31', 'SOC000028', '27');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000028', date '2023-06-01',
250, date '2023-11-30', 'SOC000029', '28');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000029', date '2023-06-01',
450, date '2023-12-31', 'SOC000030', '29');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000030', date '2023-06-01',
750, date '2025-05-31', 'SOC000031', '30');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000031', date '2023-07-01',
50, date '2023-07-31', 'SOC000032', '31');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000032', date '2023-07-01',
120, date '2023-09-30', 'SOC000033', '32');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000033', date '2023-07-01',
250, date '2023-12-31', 'SOC000034', '33');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000034', date '2023-07-01',
450, date '2023-12-31', 'SOC000035', '34');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000035', date '2023-07-01',
750, date '2025-06-30', 'SOC000036', '35');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000036', date '2023-08-01',
50, date '2023-08-31', 'SOC000037', '36');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000037', date '2023-08-01',
120, date '2023-10-31', 'SOC000038', '37');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000038', date '2023-08-01',
250, date '2023-01-31', 'SOC000039', '38');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000039', date '2023-08-01',
450, date '2023-12-31', 'SOC000040', '39');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000040', date '2023-08-01',
750, date '2025-07-31', 'SOC000042', '40');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000041', date '2023-09-01',
50, date '2023-09-30', 'SOC000043', '41');

insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000042', date '2023-09-01',
120, date '2023-11-30', 'SOC000044', '42');

```

```
insert into abbonamento(codice_abbonamento, data_inizio_abb, costo, scadenza,
codice_socio_abb, numero_armadietto_abb) values ('AB0000015', date '2023-03-01',
750, date '2025-02-28', 'SOC000016', '15');
```

POPOLAMENTO DELLA TABELLA ATTIVITA:

```
insert into attivita(codice_attivita, nome_attivita, equip_richiesto) values
('A00000001', 'Fitness', 'Tappetino');
insert into attivita(codice_attivita, nome_attivita, equip_richiesto) values
('A00000002', 'Attivita acquatiche', 'Costume');
insert into attivita(codice_attivita, nome_attivita, equip_richiesto) values
('A00000003', 'Arti marziali', 'Kimono');
insert into attivita(codice_attivita, nome_attivita, equip_richiesto) values
('A00000004', 'Combattimento', 'Paradenti');
insert into attivita(codice_attivita, nome_attivita, equip_richiesto) values
('A00000005', 'Sport di squadra', 'Scarpini');
insert into attivita(codice_attivita, nome_attivita, equip_richiesto) values
('A00000006', 'Sport individuali', 'Racchetta');
insert into attivita(codice_attivita, nome_attivita, equip_richiesto) values
('A00000007', 'Danza', 'Tutina');
```

POPOLAMENTO DELLA TABELLA INTERESSA:

```
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000001', 'SOC000001');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000002', 'SOC000001');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000003', 'SOC000002');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000004', 'SOC000002');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000005', 'SOC000004');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000006', 'SOC000004');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000007', 'SOC000005');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000001', 'SOC000005');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000002', 'SOC000008');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000003', 'SOC000008');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000004', 'SOC000009');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000005', 'SOC000009');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000006', 'SOC000010');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000007', 'SOC000010');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000001', 'SOC000007');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000002', 'SOC000007');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000003', 'SOC000011');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000004', 'SOC000011');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000005', 'SOC000012');
```

[illegible]


```

insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000001', 'SOC000042');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000002', 'SOC000043');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000003', 'SOC000043');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000004', 'SOC000044');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000005', 'SOC000044');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000006', 'SOC000006');
insert into interessa(codice_attivita_int, codice_socio_int) values
('A00000007', 'SOC000006');

```

POPOLAMENTO DELLA TABELLA CORSO:

```

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Zumba', date '2022-07-10', date '2022-08-15',
'Lunedì', to_timestamp('18:00', 'hh24:mi'), 'Rossi', 50);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Pilates', date '2022-07-15', date
'2022-08-20', 'Sabato', to_timestamp('10:30', 'hh24:mi'), 'Bianchi', 60);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Crossfit', date '2022-08-01', date
'2022-09-05', 'Lunedì', to_timestamp('19:30', 'hh24:mi'), 'Ferrari', 80);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Spinning', date '2022-08-05', date
'2022-09-09', 'Venerdì', to_timestamp('17:30', 'hh24:mi'), 'Neri', 55);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Ginnastica posturale', date '2022-09-01', date
'2022-10-06', 'Venerdì', to_timestamp('09:30', 'hh24:mi'), 'Verdi', 45);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Nuoto', date '2022-09-05', date '2022-10-10',
'Lunedì', to_timestamp('16:00', 'hh24:mi'), 'Russo', 70);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Acquagym', date '2022-10-02', date
'2022-11-06', 'Domenica', to_timestamp('11:00', 'hh24:mi'), 'Martini', 55);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Acquafitness', date '2022-10-10', date
'2022-11-14', 'Lunedì', to_timestamp('20:00', 'hh24:mi'), 'Rizzo', 60);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Idrobike', date '2022-11-03', date
'2022-12-08', 'Giovedì', to_timestamp('18:30', 'hh24:mi'), 'Esposito', 65);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Nuoto sincronizzato', date '2022-11-07', date
'2022-12-12', 'Lunedì', to_timestamp('10:00', 'hh24:mi'), 'Romano', 75);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Karate', date '2022-11-15', date '2022-12-20',
'Martedì', to_timestamp('19:30', 'hh24:mi'), 'De Luca', 70);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Judo', date '2022-12-01', date '2023-01-05',
'Giovedì', to_timestamp('17:30', 'hh24:mi'), 'Colombo', 65);

```



```

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Taekwondo', date '2022-12-05', date
'2023-01-09', 'Lunedì', to_timestamp('19:00', 'hh24:mi'), 'Costa', 70);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Boxe', date '2022-12-12', date '2023-01-16',
'Giovedì', to_timestamp('20:30', 'hh24:mi'), 'Ricci', 75);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Kickboxing', date '2022-12-15', date
'2023-01-19', 'Venerdì', to_timestamp('18:00', 'hh24:mi'), 'Cattaneo', 80);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Krav maga', date '2023-01-02', date
'2023-02-06', 'Lunedì', to_timestamp('21:00', 'hh24:mi'), 'Barbieri', 85);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Calcio', date '2023-01-10', date
'2023-02-14', 'Martedì', to_timestamp('15:00', 'hh24:mi'), 'Gatti', 50);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Pallavolo', date '2023-02-01', date
'2023-03-06', 'Lunedì', to_timestamp('20:00', 'hh24:mi'), 'Moretti', 55);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Basket', date '2023-02-05', date
'2023-03-10', 'Sabato', to_timestamp('16:00', 'hh24:mi'), 'Serra', 60);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Tennis', date '2023-02-15', date '2023-03-19',
'Mercoledì', to_timestamp('17:00', 'hh24:mi'), 'Rinaldi', 40);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Padel', date '2023-03-02', date '2023-04-06',
'Giovedì', to_timestamp('18:30', 'hh24:mi'), 'Marini', 35);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Ping pong', date '2023-03-05', date
'2023-04-09', 'Domenica', to_timestamp('12:00', 'hh24:mi'), 'De Santis', 35);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Danza classica', date '2023-03-15', date
'2023-04-19', 'Mercoledì', to_timestamp('18:00', 'hh24:mi'), 'Vitale', 30);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Danza moderna', date '2023-03-20', date
'2023-04-24', 'Lunedì', to_timestamp('19:30', 'hh24:mi'), 'Monti', 65);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Hip hop', date '2023-04-01', date
'2023-05-06', 'Sabato', to_timestamp('20:00', 'hh24:mi'), 'Ferrara', 55);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Salsa', date '2023-04-05', date '2023-05-10',
'Mercoledì', to_timestamp('21:00', 'hh24:mi'), 'Bernardi', 50);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Bachata', date '2023-04-12', date
'2023-05-17', 'Mercoledì', to_timestamp('20:30', 'hh24:mi'), 'Stella', 55);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Tango', date '2023-05-01', date '2023-06-05',
'Lunedì', to_timestamp('19:00', 'hh24:mi'), 'Romano', 60);

insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,
istruttore_corso, costo) values ('Pole dance', date '2023-05-05', date
'2023-06-09', 'Venerdì', to_timestamp('18:00', 'hh24:mi'), 'Gennaro', 70);

```

POPOLAMENTO DELLA TABELLA IMPIANTO:

```
insert into impianto(nome_impianto, copertura, tipo) values ('Michael Jordan',
'Aperto', 'Basket');
insert into impianto(nome_impianto, copertura, tipo) values ('Diego Armando
Maradona', 'Chiuso', 'Calcio');
insert into impianto(nome_impianto, copertura, tipo) values ('Roger Federer',
'Aperto', 'Tennis');
insert into impianto(nome_impianto, copertura, tipo) values ('Ivan Zaitsev',
'Aperto', 'Pallavolo');
insert into impianto(nome_impianto, copertura, tipo) values ('Francisco
Navarro', 'Chiuso', 'Padel');
insert into impianto(nome_impianto, copertura, tipo) values ('LeBron James',
'Aperto', 'Basket');
insert into impianto(nome_impianto, copertura, tipo) values ('Lionel Messi',
'Chiuso', 'Calcio');
insert into impianto(nome_impianto, copertura, tipo) values ('Rafael Nadal',
'Aperto', 'Tennis');
insert into impianto(nome_impianto, copertura, tipo) values ('Misty
May-Treanor', 'Aperto', 'Pallavolo');
insert into impianto(nome_impianto, copertura, tipo) values ('Fernando
Belasteguín', 'Chiuso', 'Padel');
insert into impianto(nome_impianto, copertura, tipo) values ('Kobe Bryant',
'Aperto', 'Basket');
insert into impianto(nome_impianto, copertura, tipo) values ('Cristiano
Ronaldo', 'Chiuso', 'Calcio');
insert into impianto(nome_impianto, copertura, tipo) values ('Novak Djokovic',
'Aperto', 'Tennis');
insert into impianto(nome_impianto, copertura, tipo) values ('Zhu Ting',
'Aperto', 'Pallavolo');
insert into impianto(nome_impianto, copertura, tipo) values ('Alejandro Galán',
'Chiuso', 'Padel');
insert into impianto(nome_impianto, copertura, tipo) values ('Mike Tyson',
'Aperto', 'Boxe');
```

POPOLAMENTO DELLA TABELLA PRENOTAZIONE:

```
insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-01-05', to_timestamp('14:30', 'hh24:mi'),
35, 'SOC000001', 'Michael Jordan');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-01-10', to_timestamp('09:00', 'hh24:mi'),
70, 'SOC000002', 'Diego Armando Maradona');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-01-15', to_timestamp('16:45', 'hh24:mi'),
25, 'SOC000004', 'Roger Federer');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-01-22', to_timestamp('11:30', 'hh24:mi'),
20, 'SOC000005', 'Ivan Zaitsev');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-02-05', to_timestamp('14:00', 'hh24:mi'),
30, 'SOC000006', 'Francisco Navarro');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-02-10', to_timestamp('10:15', 'hh24:mi'),
35, 'SOC000008', 'LeBron James');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-02-15', to_timestamp('17:30', 'hh24:mi'),
80, 'SOC000009', 'Lionel Messi');
```



```

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-02-22', to_timestamp('13:45', 'hh24:mi'),
20, 'SOC000010', 'Rafael Nadal');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-03-05', to_timestamp('11:00', 'hh24:mi'),
15, 'SOC000007', 'Misty May-Treanor');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-03-10', to_timestamp('08:15', 'hh24:mi'),
30, 'SOC000011', 'Fernando Belasteguin');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-03-15', to_timestamp('15:30', 'hh24:mi'),
35, 'SOC000012', 'Kobe Bryant');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-03-22', to_timestamp('12:45', 'hh24:mi'),
70, 'SOC000013', 'Cristiano Ronaldo');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-04-05', to_timestamp('14:30', 'hh24:mi'),
25, 'SOC000014', 'Novak Djokovic');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-04-10', to_timestamp('09:00', 'hh24:mi'),
15, 'SOC000015', 'Zhu Ting');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-04-15', to_timestamp('16:45', 'hh24:mi'),
30, 'SOC000016', 'Alejandro Galan');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-04-22', to_timestamp('11:30', 'hh24:mi'),
40, 'SOC000017', 'Mike Tyson');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-05-05', to_timestamp('14:00', 'hh24:mi'),
35, 'SOC000018', 'Michael Jordan');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-05-10', to_timestamp('10:15', 'hh24:mi'),
70, 'SOC000019', 'Diego Armando Maradona');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-05-15', to_timestamp('17:30', 'hh24:mi'),
25, 'SOC000020', 'Roger Federer');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_impianto_pren) values (date '2023-05-22', to_timestamp('13:45', 'hh24:mi'),
20, 'SOC000021', 'Ivan Zaitsev');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2023-07-07', to_timestamp('15:30', 'HH24:MI'), 60,
'SOC000001', 'Zumba');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2023-07-08', to_timestamp('11:00', 'HH24:MI'), 45,
'SOC000002', 'Pilates');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2023-07-28', to_timestamp('19:00', 'HH24:MI'), 80,
'SOC000043', 'Crossfit');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2023-08-02', to_timestamp('14:30', 'HH24:MI'), 55,
'SOC000004', 'Spinning');

```

```

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2023-08-27', to_timestamp('10:00', 'HH24:MI'), 40,
'SOC000005', 'Ginnastica posturale');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2023-09-01', to_timestamp('16:30', 'HH24:MI'), 70,
'SOC000006', 'Nuoto');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2023-10-01', to_timestamp('13:00', 'HH24:MI'), 35,
'SOC000007', 'Acquagym');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2023-10-07', to_timestamp('20:30', 'HH24:MI'), 40,
'SOC000008', 'Acquafitness');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2023-11-01', to_timestamp('17:00', 'HH24:MI'), 55,
'SOC000009', 'Idrobike');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2023-11-05', to_timestamp('13:30', 'HH24:MI'), 65,
'SOC000010', 'Nuoto sincronizzato');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2023-11-13', to_timestamp('21:00', 'HH24:MI'), 70,
'SOC000011', 'Karate');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2023-11-26', to_timestamp('17:30', 'HH24:MI'), 60,
'SOC000012', 'Judo');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2023-12-02', to_timestamp('09:30', 'HH24:MI'), 65,
'SOC000013', 'Taekwondo');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2023-12-11', to_timestamp('18:30', 'HH24:MI'), 75,
'SOC000014', 'Boxe');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2023-12-10', to_timestamp('14:00', 'HH24:MI'), 70,
'SOC000015', 'Kickboxing');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-01-01', to_timestamp('12:30', 'HH24:MI'), 80,
'SOC000016', 'Krav maga');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-01-09', to_timestamp('18:00', 'HH24:MI'), 50,
'SOC000017', 'Calcio');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-01-29', to_timestamp('15:30', 'HH24:MI'), 45,
'SOC000018', 'Pallavolo');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-02-04', to_timestamp('10:00', 'HH24:MI'), 55,
'SOC000019', 'Basket');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-02-14', to_timestamp('16:30', 'HH24:MI'), 60,
'SOC000020', 'Tennis');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-03-01', to_timestamp('13:00', 'HH24:MI'), 50,
'SOC000021', 'Padel');

```

```

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-03-03', to_timestamp('11:30', 'HH24:MI'), 40,
'SOC000022', 'Ping pong');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-03-10', to_timestamp('17:30', 'HH24:MI'), 55,
'SOC000023', 'Danza classica');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-03-28', to_timestamp('14:00', 'HH24:MI'), 65,
'SOC000024', 'Danza moderna');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-04-03', to_timestamp('20:30', 'HH24:MI'), 70,
'SOC000025', 'Hip hop');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-04-10', to_timestamp('18:00', 'HH24:MI'), 75,
'SOC000026', 'Salsa');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-04-27', to_timestamp('15:30', 'HH24:MI'), 80,
'SOC000027', 'Bachata');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-05-03', to_timestamp('19:00', 'HH24:MI'), 50,
'SOC000028', 'Tango');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-05-11', to_timestamp('21:30', 'HH24:MI'), 65,
'SOC000029', 'Pole dance');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-05-19', to_timestamp('16:00', 'HH24:MI'), 60,
'SOC000030', 'Zumba');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-05-25', to_timestamp('13:30', 'HH24:MI'), 45,
'SOC000031', 'Pilates');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-06-03', to_timestamp('10:00', 'HH24:MI'), 80,
'SOC000032', 'Crossfit');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-06-11', to_timestamp('17:30', 'HH24:MI'), 55,
'SOC000033', 'Spinning');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-06-30', to_timestamp('09:30', 'HH24:MI'), 40,
'SOC000034', 'Ginnastica posturale');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-07-04', to_timestamp('16:00', 'HH24:MI'), 70,
'SOC000035', 'Nuoto');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-07-11', to_timestamp('11:00', 'HH24:MI'), 35,
'SOC000036', 'Acquagym');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-07-18', to_timestamp('18:30', 'HH24:MI'), 40,
'SOC000037', 'Acquafitness');

insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-07-29', to_timestamp('17:00', 'HH24:MI'), 55,
'SOC000038', 'Idrobike');

```

```
insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-08-04', to_timestamp('13:30', 'HH24:MI'), 65,
'SOC000039', 'Nuoto sincronizzato');
```

```
insert into prenotazione(data_pren, ora_pren, costo_pren, codice_socio_pren,
nome_corso_pren) values (date '2024-08-10', to_timestamp('21:00', 'HH24:MI'), 70,
'SOC000040', 'Karate');
```

POPOLAMENTO DELLA TABELLA DITTA MANUTENZIONE:

```
insert into ditta_man(nome_ditta, contratto) values ('Manutec S.r.l.',
'Manutenzione preventiva');
insert into ditta_man(nome_ditta, contratto) values ('RipariFacili',
'Riparazione');
insert into ditta_man(nome_ditta, contratto) values ('Assistech', 'Assistenza
tecnica');
insert into ditta_man(nome_ditta, contratto) values ('ManuExpert',
'Manutenzione preventiva');
insert into ditta_man(nome_ditta, contratto) values ('RiparaPlus',
'Riparazione');
insert into ditta_man(nome_ditta, contratto) values ('Assistech S.p.A.',
'Assistenza tecnica');
insert into ditta_man(nome_ditta, contratto) values ('ManuPro', 'Manutenzione
preventiva');
insert into ditta_man(nome_ditta, contratto) values ('RiparaMigliore',
'Riparazione');
insert into ditta_man(nome_ditta, contratto) values ('TecnoAssist', 'Assistenza
tecnica');
insert into ditta_man(nome_ditta, contratto) values ('ManuTech', 'Manutenzione
preventiva');
```

POPOLAMENTO DELLA TABELLA MANUTENZIONE:

```
insert into manutenzione(data_inizio_man, data_fine_man, costo,
nome_impianto_man, nome_ditta_man) values (date '2023-05-15', date '2023-05-18',
200, 'Michael Jordan', 'Manutec S.r.l.');
```

```
insert into manutenzione(data_inizio_man, data_fine_man, costo,
nome_impianto_man, nome_ditta_man) values (date '2023-06-02', date '2023-06-05',
250, 'Diego Armando Maradona', 'RipariFacili');
```

```
insert into manutenzione(data_inizio_man, data_fine_man, costo,
nome_impianto_man, nome_ditta_man) values (date '2023-06-10', date '2023-06-12',
180, 'Roger Federer', 'Assistech');
```

```
insert into manutenzione(data_inizio_man, data_fine_man, costo,
nome_impianto_man, nome_ditta_man) values (date '2023-06-20', date '2023-06-22',
260, 'Ivan Zaitsev', 'ManuExpert');
```

```
insert into manutenzione(data_inizio_man, data_fine_man, costo,
nome_impianto_man, nome_ditta_man) values (date '2023-07-05', date '2023-07-07',
170, 'Francisco Navarro', 'RiparaPlus');
```

```
insert into manutenzione(data_inizio_man, data_fine_man, costo,
nome_impianto_man, nome_ditta_man) values (date '2023-07-12', date '2023-07-14',
400, 'LeBron James', 'Assistech S.p.A.');
```

```
insert into manutenzione(data_inizio_man, data_fine_man, costo,
nome_impianto_man, nome_ditta_man) values (date '2023-07-20', date '2023-07-22',
350, 'Lionel Messi', 'ManuPro');
```

```
insert into manutenzione(data_inizio_man, data_fine_man, costo,
nome_impianto_man, nome_ditta_man) values (date '2023-07-28', date '2023-07-30',
280, 'Rafael Nadal', 'RiparaMigliore');
```

POPOLAMENTO DELLA TABELLA EVENTO:

```
insert into evento(codice_evento, nome_evento, data_evento, ora_inizio_ev,
ora_fine_ev, tipo_evento, numero_spett, nome_impianto_ev) values ('EV1', 'Basket
Slam Dunk Showdown', date '2023-06-10', to_timestamp('18:00', 'hh24:mi'),
to_timestamp('20:00', 'hh24:mi'), 'Show', 500, 'Michael Jordan');

insert into evento(codice_evento, nome_evento, data_evento, ora_inizio_ev,
ora_fine_ev, tipo_evento, numero_spett, nome_impianto_ev) values ('EV2', 'Calcio
Stellare', date '2023-06-15', to_timestamp('19:30', 'hh24:mi'),
to_timestamp('22:00', 'hh24:mi'), 'Campionato', 1000, 'Diego Armando Maradona');

insert into evento(codice_evento, nome_evento, data_evento, ora_inizio_ev,
ora_fine_ev, tipo_evento, numero_spett, nome_impianto_ev) values ('EV3', 'Tennis
Swing Masters', date '2023-06-20', to_timestamp('14:00', 'hh24:mi'),
to_timestamp('17:00', 'hh24:mi'), 'Torneo', 300, 'Roger Federer');

insert into evento(codice_evento, nome_evento, data_evento, ora_inizio_ev,
ora_fine_ev, tipo_evento, numero_spett, nome_impianto_ev) values ('EV4',
'Pallavolo Sky High Tournament', date '2023-06-25', to_timestamp('16:30',
'hh24:mi'), to_timestamp('19:00', 'hh24:mi'), 'Torneo', 200, 'Ivan Zaitsev');

insert into evento(codice_evento, nome_evento, data_evento, ora_inizio_ev,
ora_fine_ev, tipo_evento, numero_spett, nome_impianto_ev) values ('EV5', 'Padel
Racket Rumble', date '2023-06-30', to_timestamp('17:30', 'hh24:mi'),
to_timestamp('20:30', 'hh24:mi'), 'Partita', 150, 'Francisco Navarro');
```

POPOLAMENTO DELLA TABELLA SPONSOR:

```
insert into sponsor(nome_sponsor, settore_sponsor, anno_fondazione) values ('ABC
Corporation', 'Tecnologia', 1990);
insert into sponsor(nome_sponsor, settore_sponsor, anno_fondazione) values
('Sportive Apparel', 'Abbigliamento sportivo', 2005);
insert into sponsor(nome_sponsor, settore_sponsor, anno_fondazione) values
('Healthy Drinks Co.', 'Alimenti e bevande', 1998);
insert into sponsor(nome_sponsor, settore_sponsor, anno_fondazione) values
('Global Financial Services', 'Servizi finanziari', 1985);
insert into sponsor(nome_sponsor, settore_sponsor, anno_fondazione) values
('Fitness Equipment Ltd.', 'Attrezzature per il fitness', 2002);
```

POPOLAMENTO DELLA TABELLA SPONSORIZZA:

```
insert into sponsorizza(codice_evento_sp, nome_sponsor_sp) values ('EV1', 'ABC
Corporation');
insert into sponsorizza(codice_evento_sp, nome_sponsor_sp) values ('EV1',
'Sportive Apparel');
insert into sponsorizza(codice_evento_sp, nome_sponsor_sp) values ('EV2',
'Healthy Drinks Co.');
```

```
insert into sponsorizza(codice_evento_sp, nome_sponsor_sp) values ('EV3', 'Global
Financial Services');
insert into sponsorizza(codice_evento_sp, nome_sponsor_sp) values ('EV4',
'Fitness Equipment Ltd.');
```

POPOLAMENTO DELLA TABELLA TURNO:

```
insert into turno (codice_turno, orario_inizio_t, orario_fine_t, tipo_t) values
('TU0001', to_timestamp('08:00', 'HH24:MI'), to_timestamp('14:00', 'HH24:MI'),
'Mattina');
insert into turno (codice_turno, orario_inizio_t, orario_fine_t, tipo_t) values
('TU0002', to_timestamp('08:00', 'HH24:MI'), to_timestamp('14:00', 'HH24:MI'),
'Mattina');
insert into turno (codice_turno, orario_inizio_t, orario_fine_t, tipo_t) values
('TU0003', to_timestamp('08:00', 'HH24:MI'), to_timestamp('14:00', 'HH24:MI'),
'Mattina');
```


[illegible]

[illegible]


```

insert into turno (codice_turno, orario_inizio_t, orario_fine_t, tipo_t) values
('TU0070', to_timestamp('09:00', 'HH24:MI'), to_timestamp('13:00', 'HH24:MI'),
'Mattina');
insert into turno (codice_turno, orario_inizio_t, orario_fine_t, tipo_t) values
('TU0071', to_timestamp('09:00', 'HH24:MI'), to_timestamp('13:00', 'HH24:MI'),
'Mattina');
insert into turno (codice_turno, orario_inizio_t, orario_fine_t, tipo_t) values
('TU0072', to_timestamp('09:00', 'HH24:MI'), to_timestamp('13:00', 'HH24:MI'),
'Mattina');
insert into turno (codice_turno, orario_inizio_t, orario_fine_t, tipo_t) values
('TU0073', to_timestamp('09:00', 'HH24:MI'), to_timestamp('13:00', 'HH24:MI'),
'Mattina');

```

POPOLAMENTO DELLA TABELLA PERSONALE:

```

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0012', 'Valentina', 'Cattanei',
'Via Trieste 55', '80210', 'Napoli', 4500, 'TU0001'); -- Medico

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0013', 'Andrea', 'Barbera', 'Via
Milano 60', '80220', 'Napoli', 3000, 'TU0004'); -- Personale amministrativo

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0014', 'Elena', 'Galli', 'Via
Roma 65', '80230', 'Napoli', 1500, 'TU0009'); -- Addetto pulizie

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0015', 'Simone', 'Morello', 'Via
Garibaldi 70', '80240', 'Napoli', 2000, 'TU0018'); -- Addetto bar

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0016', 'Greta', 'Santoro', 'Via
Verdi 75', '80250', 'Napoli', 1500, 'TU0010'); -- Addetto pulizie

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0017', 'Davide', 'Serra', 'Via
Napoli 80', '80260', 'Napoli', 2500, 'TU0005'); -- Personale amministrativo

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0018', 'Valeria', 'Rinaldi',
'Via Venezia 85', '80270', 'Napoli', 1500, 'TU0011'); -- Addetto pulizie

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0019', 'Fabio', 'Marini', 'Via
Firenze 90', '80280', 'Napoli', 2000, 'TU0019'); -- Addetto bar

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0020', 'Sofia', 'De Santis',
'Via Padova 95', '80290', 'Napoli', 1500, 'TU0012'); -- Addetto pulizie

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0021', 'Lorenzo', 'Vitale', 'Via
Genova 100', '80300', 'Napoli', 2500, 'TU0006'); -- Personale amministrativo

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0022', 'Giorgia', 'Monti', 'Via
Torino 105', '80310', 'Napoli', 1500, 'TU0013'); -- Addetto pulizie

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0023', 'Francesco', 'Ferrara',
'Via Milano 110', '80320', 'Napoli', 2000, 'TU0020'); -- Addetto bar

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0024', 'Francesca', 'Bernardi',
'Via Roma 115', '80330', 'Napoli', 1500, 'TU0014'); -- Addetto pulizie

```

```

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0025', 'Luca', 'Stella', 'Via
Garibaldi 120', '80340', 'Napoli', 2500, 'TU0007'); -- Personale
amministrativo

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0026', 'Alessia', 'Romano', 'Via
Verdi 125', '80350', 'Napoli', 1500, 'TU0015'); -- Addetto pulizie

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0027', 'Antonio', 'Gennaro',
'Via Napoli 130', '80360', 'Napoli', 2000, 'TU0021'); -- Addetto bar

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0028', 'Giulia', 'Ricci', 'Via
Venezia 135', '80370', 'Napoli', 1500, 'TU0016'); -- Addetto pulizie

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0029', 'Riccardo', 'Colombo',
'Via Firenze 140', '80380', 'Napoli', 2500, 'TU0008'); -- Personale
amministrativo

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0030', 'Elisa', 'De Luca', 'Via
Padova 145', '80390', 'Napoli', 1500, 'TU0017'); -- Addetto pulizie

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0033', 'Giovanni', 'Fabbri',
'Via Bologna 160', '80420', 'Napoli', 4800, 'TU0002'); -- Medico

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0034', 'Francesca', 'Mancini',
'Via Torino 165', '80430', 'Napoli', 5200, 'TU0003'); -- Medico

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0035', 'Luca', 'Rossi', 'Via
Verona 170', '80440', 'Napoli', 3000, 'TU0045'); -- Istruttore zumba

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0036', 'Simona', 'Bianchi', 'Via
Roma 175', '80450', 'Napoli', 2800, 'TU0046'); -- Istruttrice pilates

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0037', 'Fabio', 'Ferrari', 'Via
Firenze 180', '80460', 'Napoli', 3200, 'TU0047'); -- Istruttore crossfit

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0038', 'Valentina', 'Neri', 'Via
Milano 185', '80470', 'Napoli', 2900, 'TU0048'); -- Istruttrice spinning

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0039', 'Alessio', 'Verdi', 'Via
Torino 190', '80480', 'Napoli', 3100, 'TU0049'); -- Istruttore ginnastica

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0040', 'Martina', 'Russo', 'Via
Napoli 195', '80490', 'Napoli', 3000, 'TU0050'); -- Istruttrice nuoto

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0041', 'Roberto', 'Martini',
'Via Salerno 200', '80500', 'Napoli', 2800, 'TU0051'); -- Istruttore acquagym

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0042', 'Maria', 'Rizzo', 'Via
Catania 205', '80510', 'Napoli', 3200, 'TU0052'); -- Istruttrice acquafitness

```

```

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0043', 'Paolo', 'Esposito', 'Via
Messina 210', '80520', 'Napoli', 2900, 'TU0053'); -- Istruttore idrobike

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0044', 'Laura', 'Romano', 'Via
Palermo 215', '80530', 'Napoli', 3100, 'TU0054'); -- Istruttrice nuoto
sincronizzato

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0045', 'Antonio', 'De Luca',
'Via Padova 220', '80540', 'Napoli', 3000, 'TU0055'); -- Istruttore karate

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0046', 'Silvia', 'Colombo', 'Via
Genova 225', '80550', 'Napoli', 2800, 'TU0056'); -- Istruttrice judo

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0047', 'Giuseppe', 'Costa', 'Via
Venezia 230', '80560', 'Napoli', 3200, 'TU0057'); -- Istruttore taekwondo

```

```

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0048', 'Francesca', 'Ricci',
'Via Firenze 235', '80570', 'Napoli', 2900, 'TU0058'); -- Istruttrice boxe

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0049', 'Luigi', 'Cattaneo', 'Via
Milano 240', '80580', 'Napoli', 3100, 'TU0059'); -- Istruttore kickboxing

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0050', 'Francesca', 'Barbieri',
'Via Torino 245', '80590', 'Napoli', 3000, 'TU0060'); -- Istruttrice krav
maga

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0051', 'Marco', 'Gatti', 'Via
Napoli 250', '80600', 'Napoli', 2800, 'TU0061'); -- Istruttore calcio

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0052', 'Alessandra', 'Moretti',
'Via Salerno 255', '80610', 'Napoli', 3200, 'TU0062'); -- Istruttrice
pallavolo

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0011', 'Luca', 'Serra', 'Via
Catania 260', '80620', 'Napoli', 2900, 'TU0063'); -- Istruttore basket

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0010', 'Sara', 'Rinaldi', 'Via
Messina 265', '80630', 'Napoli', 3100, 'TU0064'); -- Istruttrice tennis

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0009', 'Giovanni', 'Marini',
'Via Palermo 270', '80640', 'Napoli', 3000, 'TU0065'); -- Istruttore padel

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0008', 'Francesca', 'De Santis',
'Via Padova 275', '80650', 'Napoli', 2800, 'TU0066'); -- Istruttrice ping
pong

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0007', 'Marco', 'Vitale', 'Via
Genova 280', '80660', 'Napoli', 3200, 'TU0067'); -- Istruttore danza classica

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0006', 'Chiara', 'Monti', 'Via
Venezia 285', '80670', 'Napoli', 2900, 'TU0068'); -- Istruttrice danza
moderna

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0005', 'Davide', 'Ferrara', 'Via
Firenze 290', '80680', 'Napoli', 3100, 'TU0069'); -- Istruttore hip hop

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0004', 'Maria', 'Bernardi',
'Via Milano 295', '80690', 'Napoli', 3000, 'TU0070'); -- Istruttrice salsa

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0004', 'Maria', 'Bernardi', 'Via
Milano 295', '80690', 'Napoli', 3000, 'TU0070'); -- Istruttrice salsa

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0004', 'Maria', 'Bernardi',
'Via Milano 295', '80690', 'Napoli', 3000, 'TU0070'); -- Istruttrice salsa

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0003', 'Luigi', 'Stella', 'Via
Torino 300', '80700', 'Napoli', 2800, 'TU0071'); -- Istruttore bachata

```

```

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0004', 'Maria', 'Bernardi',
'Via Milano 295', '80690', 'Napoli', 3000, 'TU0070'); -- Istruttrice salsa
insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0002', 'Valentina', 'Romano',
'Via Napoli 305', '80710', 'Napoli', 3200, 'TU0072'); -- Istruttrice tango

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0004', 'Maria', 'Bernardi',
'Via Milano 295', '80690', 'Napoli', 3000, 'TU0070'); -- Istruttrice salsa
insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0001', 'Michele', 'Gennaro',
'Via Salerno 310', '80720', 'Napoli', 2900, 'TU0073'); -- Istruttore pole dance

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0004', 'Maria', 'Bernardi',
'Via Milano 295', '80690', 'Napoli', 3000, 'TU0070'); -- Istruttrice salsa
insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0053', 'Luigi', 'Bianchi', 'Via
Manzoni 234', '80147', 'Napoli', 1800, 'TU0022'); -- Addetto bar

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0054', 'Sara', 'Ferrari', 'Via
Volta 567', '80148', 'Napoli', 1600, 'TU0023'); -- Addetto bar

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0055', 'Mario', 'Costa', 'Via
Garibaldi 123', '80141', 'Napoli', 1800, 'TU0024'); -- Addetto bar

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0056', 'Anna', 'Ricci', 'Via
Roma 456', '80142', 'Napoli', 1600, 'TU0025'); -- Addetto bar

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0057', 'Giovanni', 'Cattaneo',
'Via Verdi 789', '80143', 'Napoli', 1800, 'TU0026'); -- Addetto bar

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0058', 'Laura', 'Barbieri', 'Via
Mazzini 321', '80144', 'Napoli', 1600, 'TU0027'); -- Addetto bar

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0059', 'Antonio', 'Gatti', 'Via
Dante 654', '80145', 'Napoli', 1800, 'TU0028'); -- Addetto bar

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0060', 'Francesca', 'Moretti',
'Via Leopardi 987', '80146', 'Napoli', 1600, 'TU0029'); -- Addetto bar

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0061', 'Marco', 'Bianchini',
'Via Verdi 123', '80140', 'Napoli', 1600, 'TU0030'); -- Addetto bar

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0062', 'Silvia', 'Ferro', 'Via
Rossini 456', '80141', 'Napoli', 1600, 'TU0031'); -- Addetto bar

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0063', 'Roberto', 'Costanzo',
'Via Mozart 789', '80142', 'Napoli', 1600, 'TU0032'); -- Addetto bar

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0064', 'Giulia', 'Ricciardi',
'Via Verdi 123', '80143', 'Napoli', 1600, 'TU0033'); -- Addetto bar

```

```

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0065', 'Fabio', 'Marchetti',
'Via Rossini 456', '80144', 'Napoli', 1600, 'TU0034'); -- Addetto bar

insert into personale(codice_imp, nome_imp, cognome_imp, via_imp, cap_imp,
citta_imp, stipendio, codice_turno_p) values ('CI0066', 'Valeria', 'Romagnoli',
'Via Mozart 789', '80145', 'Napoli', 1600, 'TU0035'); -- Addetto bar

```

POPOLAMENTO DELLA TABELLA ISTRUTTORE:

```

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0035', 'Fitness', '28', 'Zumba');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0036', 'Fitness', '34', 'Pilates');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0037', 'Fitness', '31', 'Crossfit');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0038', 'Fitness', '29', 'Spinning');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0039', 'Fitness', '26', 'Ginnastica posturale');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0040', 'Nuoto', '33', 'Nuoto');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0041', 'Nuoto', '30', 'Acquagym');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0042', 'Nuoto', '27', 'Acquafitness');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0043', 'Fitness', '28', 'Idrobike');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0044', 'Nuoto', '32', 'Nuoto sincronizzato');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0045', 'Arti marziali', '29', 'Karate');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0046', 'Arti marziali', '30', 'Judo');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0047', 'Arti marziali', '28', 'Taekwondo');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0048', 'Arti marziali', '27', 'Boxe');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0049', 'Arti marziali', '31', 'Kickboxing');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0050', 'Arti marziali', '32', 'Krav maga');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0051', 'Calcio', '29', 'Calcio');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0052', 'Pallavolo', '26', 'Pallavolo');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0011', 'Basket', '30', 'Basket');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0010', 'Tennis', '28', 'Tennis');

```



```

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0009', 'Paddle', '33', 'Padel');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0008', 'Tennis', '29', 'Ping pong');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0007', 'Danza', '27', 'Danza classica');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0006', 'Danza', '28', 'Danza moderna');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0005', 'Danza', '30', 'Hip hop');
insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0004', 'Danza', '32', 'Salsa');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0003', 'Danza', '26', 'Bachata');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0002', 'Danza', '31', 'Tango');

insert into istruttore(codice_imp_istr, sport_praticato, eta_istruttore,
nome_corso_istr) values ('CI0001', 'Fitness', '29', 'Pole dance');

POPOLAMENTO DELLA TABELLA MEDICO:
insert into medico(codice_imp_med, specializzazione, anno_abilit) values
('CI0012', 'Medicina dello Sport', '2018');
insert into medico(codice_imp_med, specializzazione, anno_abilit) values
('CI0033', 'Ortopedia e Traumatologia', '2016');
insert into medico(codice_imp_med, specializzazione, anno_abilit) values
('CI0034', 'Nutrizione e Dietetica', '2017');

POPOLAMENTO DELLA TABELLA AMMINISTRATIVO:
insert into amministrativo(codice_imp_amm) values ('CI0013');
insert into amministrativo(codice_imp_amm) values ('CI0017');
insert into amministrativo(codice_imp_amm) values ('CI0021');
insert into amministrativo(codice_imp_amm) values ('CI0025');
insert into amministrativo(codice_imp_amm) values ('CI0029');

POPOLAMENTO DELLA TABELLA ADDETTO PULIZIE:
insert into addetto_pulizie(codice_imp_add) values ('CI0016');
insert into addetto_pulizie(codice_imp_add) values ('CI0018');
insert into addetto_pulizie(codice_imp_add) values ('CI0020');
insert into addetto_pulizie(codice_imp_add) values ('CI0022');
insert into addetto_pulizie(codice_imp_add) values ('CI0024');
insert into addetto_pulizie(codice_imp_add) values ('CI0026');
insert into addetto_pulizie(codice_imp_add) values ('CI0028');
insert into addetto_pulizie(codice_imp_add) values ('CI0030');

POPOLAMENTO DELLA TABELLA ADDETTO BAR:
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0015', 'Barista');
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0019', 'Responsabile
bar');
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0023', 'Cassiere');
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0027', 'Cameriere');
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0053', 'Cassiere');
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0054', 'Barista');
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0055', 'Responsabile
bar');
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0056', 'Cameriere');
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0057', 'Cassiere');

```

```

insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0058', 'Barista');
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0059', 'Responsabile
bar');
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0060', 'Cameriere');
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0061', 'Cassiere');
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0062', 'Barista');
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0063', 'Responsabile
bar');
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0064', 'Cameriere');
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0065', 'Cassiere');
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0066', 'Barista');
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0051', 'Responsabile
bar');
insert into addetto_bar(codice_imp_addbar, ruolo) values ('CI0052', 'Cameriere');

```

POPOLAMENTO DELLA TABELLA VISITA MEDICA:

```

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM01', 'Esito positivo', date '2023-01-01', 'CI0012',
'SOC000001');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM02', 'Esito positivo', date '2023-01-01', 'CI0033',
'SOC000002');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM03', 'Esito positivo', date '2023-01-01', 'CI0034',
'SOC000004');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM04', 'Esito positivo', date '2023-01-01', 'CI0012',
'SOC000005');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM05', 'Esito positivo', date '2023-01-01', 'CI0033',
'SOC000006');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM06', 'Esito positivo', date '2023-02-01', 'CI0034',
'SOC000008');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM07', 'Esito positivo', date '2023-02-01', 'CI0012',
'SOC000009');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM08', 'Esito positivo', date '2023-02-01', 'CI0033',
'SOC000010');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM09', 'Esito positivo', date '2023-02-01', 'CI0034',
'SOC000007');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM10', 'Esito positivo', date '2023-02-01', 'CI0012',
'SOC000011');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM11', 'Esito positivo', date '2023-03-01', 'CI0033',
'SOC000012');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM12', 'Esito positivo', date '2023-03-01', 'CI0034',
'SOC000013');

```


[illegible]

```

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM30', 'Esito positivo', date '2023-06-01', 'CI0034',
'SOC000031');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM31', 'Esito positivo', date '2023-07-01', 'CI0012',
'SOC000032');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM32', 'Esito positivo', date '2023-07-01', 'CI0033',
'SOC000033');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM33', 'Esito positivo', date '2023-07-01', 'CI0034',
'SOC000034');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM34', 'Esito positivo', date '2023-07-01', 'CI0012',
'SOC000035');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM35', 'Esito positivo', date '2023-07-01', 'CI0033',
'SOC000036');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM36', 'Esito positivo', date '2023-08-01', 'CI0034',
'SOC000037');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM37', 'Esito positivo', date '2023-08-01', 'CI0012',
'SOC000038');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM38', 'Esito positivo', date '2023-08-01', 'CI0033',
'SOC000039');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM40', 'Esito positivo', date '2023-08-01', 'CI0012',
'SOC000042');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM41', 'Esito positivo', date '2023-09-01', 'CI0033',
'SOC000043');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM42', 'Esito positivo', date '2023-09-01', 'CI0034',
'SOC000044');

insert into visita_medica(codice_visita, esito, data_visita, codice_imp_med_vis,
codice_socio_vis) values ('VM39', 'Esito positivo', date '2023-08-01', 'CI0034',
'SOC000040');

```

POPOLAMENTO DELLA TABELLA MAGAZZINO:

```

insert into magazzino(codice_magazzino, telefono, via_m, cap_m, citta_m) values
('1', '0811234567', 'Via Toledo 1', '80100', 'Napoli');
insert into magazzino(codice_magazzino, telefono, via_m, cap_m, citta_m) values
('2', '0812345678', 'Via dei Tribunali 2', '80100', 'Napoli');
insert into magazzino(codice_magazzino, telefono, via_m, cap_m, citta_m) values
('3', '0813456789', 'Via San Gregorio Armeno 3', '80100', 'Napoli');
insert into magazzino(codice_magazzino, telefono, via_m, cap_m, citta_m) values
('4', '0814567890', 'Via Caracciolo 4', '80100', 'Napoli');
insert into magazzino(codice_magazzino, telefono, via_m, cap_m, citta_m) values
('5', '0815678901', 'Via Chiaia 5', '80100', 'Napoli');

```

POPOLAMENTO DELLA TABELLA ASSEGNA:

```
insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-07-01', date '2023-07-31', 'non disponibile', '1', 'AB0000001');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-07-02', date '2023-08-01', 'non disponibile', '2', 'AB0000002');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-07-03', date '2023-08-02', 'non disponibile', '3', 'AB0000003');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-07-04', date '2023-08-03', 'non disponibile', '4', 'AB0000004');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-07-05', date '2023-08-04', 'non disponibile', '5', 'AB0000005');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-07-06', date '2023-08-05', 'non disponibile', '6', 'AB0000006');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-07-07', date '2023-08-06', 'non disponibile', '7', 'AB0000007');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-07-08', date '2023-08-07', 'non disponibile', '8', 'AB0000008');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-07-09', date '2023-08-08', 'non disponibile', '9', 'AB0000009');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-07-10', date '2023-08-09', 'non disponibile', '10', 'AB0000010');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-07-11', date '2023-08-10', 'non disponibile', '11', 'AB0000011');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-07-12', date '2023-08-11', 'non disponibile', '12', 'AB0000012');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-07-13', date '2023-08-12', 'non disponibile', '13', 'AB0000013');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-07-14', date '2023-08-13', 'non disponibile', '14', 'AB0000014');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-07-15', date '2023-08-14', 'non disponibile', '15', 'AB0000015');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-07-16', date '2023-08-15', 'non disponibile', '16', 'AB0000016');
```

[illegible]

```

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-08-03', date '2023-09-02', 'non disponibile', '34', 'AB0000034');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-08-04', date '2023-09-03', 'non disponibile', '35', 'AB0000035');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-08-05', date '2023-09-04', 'non disponibile', '36', 'AB0000036');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-08-06', date '2023-09-05', 'non disponibile', '37', 'AB0000037');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-08-07', date '2023-09-06', 'non disponibile', '38', 'AB0000038');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-08-08', date '2023-09-07', 'non disponibile', '39', 'AB0000039');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-08-09', date '2023-09-08', 'non disponibile', '40', 'AB0000040');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-08-10', date '2023-09-09', 'non disponibile', '41', 'AB0000041');

insert into assegna(data_assegnazione, scadenza_assegnazione, stato,
numero_armadietto_assegna, codice_abbonamento_assegna) values (date
'2023-08-11', date '2023-09-10', 'non disponibile', '42', 'AB0000042');

```

2.4 Triggers

TRIGGER PER CONTROLLARE CHE UN SOCIO NON POSSA AVERE PIU' DI UN ABBONAMENTO ATTIVO ALLA VOLTA:

```
create or replace trigger controllo_abbonamenti_attivi
before insert on abbonamento
for each row
declare
    abbonamenti_attivi number;
begin
    select count(*) into abbonamenti_attivi
    from abbonamento
    where codice_socio_abb = :new.codice_socio_abb and scadenza >= sysdate;
    if abbonamenti_attivi > 0 then raise_application_error(-20001, 'Il socio
        ha già un abbonamento attivo.');
```

TRIGGER PER CONTROLLARE SE IL NUMERO DEI SOCI SUPERI IL NUMERO DI ARMADIETTI DISPONIBILI:

```
create or replace trigger controllo_numero_soci_armadietti
before insert or update on abbonamento
for each row
declare
    num_soci number;
    num_armadietti number;
begin
    select count(*) into num_soci
    from socio;
    select count(*) into num_armadietti
    from armadietto;
    if num_soci > num_armadietti then raise_application_error(-20002, 'Il
        numero di soci supera il numero di armadietti disponibili.');
```

TRIGGER PER CONTROLLARE SE IL TIPO DI IMPIANTO CORRISPONDE AD UN ALTRO TIPO INSERITO:

```
create or replace trigger controllo_tipo_impianto
before insert or update on impianto
for each row
begin
    if lower (:new.tipo) not in ('tennis', 'calcio', 'pallavolo', 'basket',
        'padel', 'boxe') then raise_application_error(-20003, 'Il tipo di
        impianto specificato non è valido. I tipi validi sono: tennis, calcio,
        pallavolo, basket, padel e boxe.');
```

TRIGGER PER CONTROLLARE IL TIPO DI CONTRATTO:

```
create or replace trigger controllo_contratto_ditta_man
before insert or update on ditta_man
for each row
begin
    if lower (:new.contratto) not in ('manutenzione preventiva',
        'riparazione', 'assistenza tecnica') then raise_application_error(-20004,
        'Il tipo di contratto specificato non è valido. I tipi validi sono:
        manutenzione preventiva, riparazione, assistenza tecnica.');
```

TRIGGER PER CONTROLLARE LO SPORT PRATICATO:

```
create or replace trigger controllo_sport_praticato
before insert or update on istruttore
for each row
begin
    if lower (:new.sport_praticato) not in ('fitness', 'nuoto', 'arti
marziali', 'basket', 'calcio', 'tennis', 'padel', 'pallavolo', 'danza')
    then raise_application_error(-20005, 'Lo sport praticato specificato non
è valido. Gli sport validi sono: fitness, nuoto, arti marziali,
basket, calcio, tennis, padel, pallavolo, danza.');
```

TRIGGER PER CONTROLLARE CHE LA DATA DI INIZIO DI UN CORSO SIA PRECEDENTE ALLA DATA DI FINE:

```
create or replace trigger controllo_date_corso
before insert or update on corso
for each row
begin
    if :new.data_inizio_c >= :new.data_fine_c then
        raise_application_error(-20006, 'La data di inizio deve essere precedente
alla data di fine del corso.');
```

TRIGGER PER CONTROLLARE CHE UN SOCIO NON POSSA AVERE PIU' DI UNA VISITA MEDICA AL GIORNO:

```
create or replace trigger controllo_visite_mediche_giornaliere
before insert or update on visita_medica
for each row
declare
    visite_giornaliere number;
begin
    select count(*) into visite_giornaliere
    from visita_medica
    where codice_socio_vis = :new.codice_socio_vis and data_visita =
:new.data_visita;
    if visite_giornaliere > 0 then raise_application_error(-20007, 'Il socio
ha già effettuato una visita medica in questa data.');
```

TRIGGER PER CONTROLLARE CHE LA DATA DI INIZIO DELLA MANUTENZIONE SIA PRECEDENTE ALLA DATA DI FINE:

```
create or replace trigger controllo_data_manutenzione
before insert or update on manutenzione
for each row
begin
    if :new.data_inizio_man >= :new.data_fine_man then
        raise_application_error(-20009, 'La data di inizio della manutenzione
deve essere precedente alla data di fine.');
```

TRIGGER PER VERIFICARE CHE LA DATA DI INIZIO DELL'ABBONAMENTO SIA PRECEDENTE ALLA DATA DI SCADENZA:

```
create or replace trigger controllo_data_abbonamento
before insert or update on abbonamento
for each row
begin
    if :new.data_inizio_abb >= :new.scadenza then
        raise_application_error(-20010, 'La data di inizio dell''abbonamento deve
        essere precedente alla data di scadenza.');
```

```
    end if;
end;
```

2.5 Procedure

1) PROCEDURA CHE PERMETTE DI CREARE UN NUOVO EVENTO:

```
create or replace procedure CreaEvento (
    codice_evento_in char,
    nome_evento_in varchar,
    data_evento_in date,
    ora_inizio_ev_in timestamp,
    ora_fine_ev_in timestamp,
    tipo_evento_in varchar,
    numero_spett_in number,
    nome_impianto_ev_in varchar)
is
begin
    insert into evento(codice_evento, nome_evento, data_evento, ora_inizio_ev,
    ora_fine_ev, tipo_evento, numero_spett, nome_impianto_ev) values
    (codice_evento_in, nome_evento_in, data_evento_in, ora_inizio_ev_in,
    ora_fine_ev_in, tipo_evento_in, numero_spett_in, nome_impianto_ev_in);
    dbms_output.put_line('Evento creato con successo.');
```

```
exception
```

```
    when others then
```

```
        dbms_output.put_line('Errore durante la creazione dell'evento: ' || '20020');
```

```
end;
```

2) PROCEDURA CHE PERMETTE DI AGGIORNARE I TURNI DEL PERSONALE:

```
create or replace procedure AggiornaTurno (
    codice_turno_in char,
    orario_inizio_t_in timestamp,
    orario_fine_t_in timestamp,
    tipo_t_in varchar)
is
begin
    update turno
    set orario_inizio_t = orario_inizio_t_in, orario_fine_t = orario_fine_t_in,
    tipo_t = tipo_t_in
    where codice_turno = codice_turno_in;
    dbms_output.put_line('Turno aggiornato con successo.');
```

```
exception
```

```
    when no_data_found then
```

```
        dbms_output.put_line('Nessun turno trovato con il codice
        specificato.');
```

```
    when others then
```

```
        dbms_output.put_line('Errore durante l'aggiornamento del turno: ' ||
        '20021');
```

```
end;
```


3) PROCEDURA CHE PERMETTE DI AGGIUNGERE UN NUOVO CORSO ED ASSEGNARE IL SUO ISTRUTTORE:

```
create or replace procedure aggiungi_corso (  
    p_nome_corso corso.nome_corso%type,  
    p_data_inizio_c corso.data_inizio_c%type,  
    p_data_fine_c corso.data_fine_c%type,  
    p_giorno corso.giorno%type,  
    p_ora corso.ora%type,  
    p_istruttore_corso istruttore.codice_imp_istr%type,  
    p_costo corso.costo%type)  
is  
begin  
    insert into corso(nome_corso, data_inizio_c, data_fine_c, giorno, ora,  
        istruttore_corso, costo) values (p_nome_corso, p_data_inizio_c,  
        p_data_fine_c, p_giorno, p_ora, p_istruttore_corso, p_costo);  
    update istruttore  
    set nome_corso_istr = p_nome_corso  
    where codice_imp_istr = p_istruttore_corso;  
    commit;  
    dbms_output.put_line('Nuovo corso aggiunto con successo.');
```

exception
 when others then
 rollback;
 dbms_output.put_line('Errore durante l'aggiunta del corso: ' ||
 '20022');

```
end;
```

4) PROCEDURA CHE PERMETTE DI CANCELLARE UNA PRENOTAZIONE:

```
create or replace procedure cancella_prenotazione (  
    data_prenotazione date,  
    ora_prenotazione timestamp)  
is  
begin  
    delete from prenotazione  
    where data_pren = data_prenotazione and ora_pren = ora_prenotazione;  
    if SQL%ROWCOUNT = 0 then  
        dbms_output.put_line('Nessuna prenotazione trovata con il codice  
            specificato, data e ora.');
```

else
 dbms_output.put_line('Prenotazione cancellata con successo.');

```
    end if;  
exception  
    when others then  
        dbms_output.put_line('Si è verificato un errore durante la  
            cancellazione della prenotazione: ') || '20022');
```

```
end;
```

5) PROCEDURA CHE AGGIORNA LE PRENOTAZIONI:

```
create or replace procedure aggiorna_prenotazione (  
    data_pren date,  
    ora_pren timestamp,  
    nuovo_costo_pren number,  
    nuovo_nome_impianto_pren varchar,  
    nuovo_nome_corso_pren varchar)  
is  
begin  
    update prenotazione  
    set costo_pren = nuovo_costo_pren, nome_impianto_pren =  
    nuovo_nome_impianto_pren, nome_corso_pren = nuovo_nome_corso_pren  
    where data_pren = data_pren and ora_pren = ora_pren;  
    if SQL%ROWCOUNT = 0 then  
        dbms_output.put_line('Nessuna prenotazione trovata con la data e  
        l'ora specificata.');
```

```
    else  
        dbms_output.put_line('Prenotazione aggiornata con successo.');
```

```
    end if;  
exception  
    when others then  
        dbms_output.put_line('Si è verificato un errore durante  
        l'aggiornamento della prenotazione: ' || '20023');    end;
```

6) PROCEDURA CHE PERMETTE DI AGGIUNGERE UNA NUOVA DITTA DI MANUTENZIONE:

```
create or replace procedure aggiungi_ditta_manutenzione (  
    v_nome_ditta varchar,  
    v_contratto varchar)  
is  
begin  
    insert into ditta_man(nome_ditta, contratto) values (v_nome_ditta,  
    v_contratto);  
    dbms_output.put_line('Nuova ditta di manutenzione aggiunta con  
    successo.');
```

```
    commit;  
exception  
    when other then  
        dbms_output.put_line('Si è verificato un errore durante l'aggiunta  
        della ditta di manutenzione: ' || '20024');
```

```
    rollback;  
end;
```