



Università degli Studi di Salerno



Dipartimento di Ingegneria dell'Informazione ed Elettrica e
Matematica Applicata

Corso di Laurea in Ingegneria Informatica

Basi di Dati 2020/2021 Canale I-Z

Project Work
SportClub

Gruppo n. **16 – IZ**

WP	Cognome e Nome	Matricola	e-mail	Responsabile
1	Savarese Marco	0612704404	m.savarese18@studenti.unisa.it	
2	Landi Thomas	0612704353	t.landi3@studenti.unisa.it	
3	Matteo Memoli	0612704437	m.memoli50@studenti.unisa.it	
4	Paolillo Alessandro	0612704415	a.paolillo26@studenti.unisa.it	X

Anno accademico 2020-2021

Sommario

1. Descrizione della realtà di interesse	3
2. Analisi delle specifiche	4
2.1. Glossario dei termini	4
2.2. Strutturazione dei requisiti in frasi	4
2.2.1. Frasi di carattere generale.....	4
2.2.2. Frasi relative a utente	4
2.2.3. Frasi relative a società.....	4
2.2.4. Frasi relative a servizio	4
2.2.5. Frasi relative a invito	5
2.2.6. Frasi relative per acquisto	5
2.2.7. Frasi relative a professionista.....	5
2.2.8. Frasi relative a struttura	5
2.3. Identificazione delle operazioni principali	5
3. Progettazione Concettuale	6
3.1. Schema Concettuale	6
3.1.1. Note sullo schema E-R.....	7
3.2. Design Pattern.....	7
3.2.1. Pattern evoluzione di un concetto.....	7
3.2.2. Pattern reificazione di un attributo (disciplina)	8
3.2.3. Pattern reificazione di un attributo (account).....	9
3.3. Dizionario dei Dati	10
3.4. Regole Aziendali.....	12
4. Progettazione Logica.....	13
4.1. Ristrutturazione Schema Concettuale	13
4.1.1. Analisi delle Prestazioni.....	13
4.2. Analisi delle ridondanze.....	14
4.2.1. Analisi della ridondanza 1: Numero acquisti	15
4.2.1.1. Valutazione della ridondanza 1: Numero Acquisti	16
4.2.2. Analisi della ridondanza 2: Voga	16
4.2.2.1. Valutazione della ridondanza 2: Voga	17
4.3. Eliminazione delle generalizzazioni	17
4.3.1. Generalizzazione <i>Fornitore</i>	17
4.3.2. Generalizzazione <i>Invito</i>	18
4.4. Partizionamento/Accorpamento Entità e Associazioni	18
4.5. Scelta degli identificatori principali	19
4.6. Schema ristrutturato finale	20
4.7. Schema logico	21
4.8. Documentazione dello schema logico.....	22
5. Normalizzazione.....	23
5.1. Analisi delle dipendenze funzionali	23
5.1.1. Preferenza	23
5.1.2. Società	23
5.1.3. Professionista	23

5.1.4.	Account.....	23
5.1.5.	Proposta Società	23
5.1.6.	Proposta professionista.....	24
5.1.7.	Utilizzo.....	24
5.1.8.	Disciplina	24
5.1.9.	Struttura	24
5.1.10.	Cooperazione	24
5.1.11.	Servizio	24
5.1.12.	Acquisto.....	25
5.1.13.	Utente	25
5.1.14.	Cellulare.....	25
5.1.15.	Pagamento.....	25
5.1.16.	Invito	25
6.	Script Creazione e Popolamento Database	26
6.1.	Script di Creazione del Database	26
6.1.1.	Script di Drop	26
6.1.2.	Creazione Domini.....	26
6.1.3.	Creazione Tabelle.....	26
6.2.	Script di Popolamento del Database	30
7.	Query SQL	38
7.1.	Query con operatore di aggregazione e join: Elenco riassuntivo dei servizi offerti	38
7.2.	Query nidificata complessa: Utenti Invitati.....	38
7.3.	Query insiemistica: Tutte le società che non offrono il calcio a 11	39
7.4.	Altre Query	39
7.4.1.	Società che offrono almeno 3 servizi ma nessuno di calcio.....	39
8.	Viste.....	40
8.1.	Vista Voga.....	40
8.1.1.	Query con Vista: Voga	40
8.2.	Vista NumeroAcquisti	40
8.2.1.	Query con Vista: NumeroAcquisti	40
8.3.	Vista: Professionisti Cooperanti	41
8.3.1.	Query con vista: Cooperazioni professionista Morriello	41
9.	Trigger	42
9.1.	Trigger inizializzazione: <i>Controllo Username</i>	42
9.1.1.	Creazione trigger controllo username_utente.....	42
9.1.2.	Creazione trigger controllo username_professionista	42
9.1.3.	Creazione trigger username_società	43
9.1.4.	Creazione trigger Inserimenti in PropostaSocietà dato inserimenti in Cooperazione	43
9.2.	Trigger per vincoli aziendali	44
9.2.1.	Trigger1: Creazione trigger controllo acquisto.....	44
9.2.2.	Trigger2: Creazione trigger controllo invito condivisibile.....	45

1. Descrizione della realtà di interesse

Titolo: **SportClub**

SportClub è una piattaforma realizzata da una startup italiana dedicata a sportivi e a strutture, club, federazioni attive nel settore. Consente agli sportivi di cercare, prenotare e pagare corsi e servizi offerte dalle diverse strutture, consentendo anche di invitare i propri amici a condividere le stesse passioni. La start-up ha richiesto la creazione della base di dati per supportare tale piattaforma.

- Per gli utenti della piattaforma si intende memorizzare le informazioni anagrafiche, l'account (username e password), i recapiti, i metodi di pagamento usati, le discipline preferite, il livello di attività/esperienza per ogni disciplina (e.g., novizio, amatoriale, agonistico, etc.)
- Esistono diverse società (e.g., palestre, club, federazioni, etc.) e professionisti che possono offrire servizi e/o strutture. Ad esempio, possono offrire corsi, lezioni, ma anche la possibilità di prenotare campi da calcio, da tennis, etc. Si intendono memorizzare le informazioni su professionisti e società, incluse le discipline nelle quali offrono i propri servizi.
- I servizi possono essere: corso; lezione una-tantum; prenotazione struttura; consulenza.
- Per ogni servizio si vogliono memorizzare il titolo, la descrizione, la durata, il costo, la disciplina, il professionista eventualmente coinvolto, la struttura eventualmente coinvolta.
- Un servizio può essere acquistato da un utente; si intendono memorizzare le informazioni sugli acquisti degli utenti.
- Per alcuni servizi, è possibile invitare i propri amici (altri utenti della piattaforma). Ad esempio, è possibile prenotare un campo da calcio e invitare i propri amici per organizzare una partita. Per l'invito si vuole memorizzare l'utente che l'ha creato, la descrizione dell'evento che si sta organizzando, il servizio coinvolto, gli invitati.
- Un invitato può accettare o meno l'invito. Si vuole tenere traccia dell'invito accettato o meno, inclusa la data di accettazione o rifiuto.

2. Analisi delle specifiche

Workpackage	Task	Responsabile
WP0	Analisi delle specifiche	Intero Gruppo

2.1. Glossario dei termini

	Termine	Descrizione	Sinonimi	Collegamenti
1	Utente	Persona che usufruisce dei servizi offerti dalla società.	Invitato, amici, sportivo	Invito, Acquisto
2	Società	Organizzazione sportiva che offre servizi e/o strutture	Club, federazioni	Servizio, Struttura
3	Servizio	I servizi possono essere: corso; lezione una-tantum; prenotazione struttura; consulenza, offerti dal professionista o dalla società	-	Società, Professionista, Acquisto, Struttura
4	Invito	Richiesta di partecipazione ad un evento che deve essere tracciato	-	Utente
5	Acquisto	Il servizio che viene acquistato da un utente	Prenotazione	Utente, Servizio
6	Professionista	Persona specializzata che offre servizi e/o strutture	-	Servizio, Struttura
7	Struttura	Struttura messa a disposizione dal professionista o dalla società per l'utente	-	Servizio, Professionista, Società

2.2. Strutturazione dei requisiti in frasi

2.2.1. Frasi di carattere generale

SportClub è una piattaforma realizzata da una startup italiana dedicata a sportivi e a strutture, club, federazioni attive nel settore e consente agli sportivi di cercare, prenotare e pagare corsi e servizi offerte dalle diverse strutture, di invitare i propri amici a condividere le stesse passioni.

2.2.2. Frasi relative a utente

Per gli utenti della piattaforma rappresentiamo le informazioni anagrafiche, l'account (username e password), i recapiti, i metodi di pagamento usati, le discipline preferite, il livello di attività/esperienza per ogni disciplina (e.g., novizio, amatoriale, agonistico, etc.).

2.2.3. Frasi relative a società

Per le società (e.g., palestre, club, federazioni, etc.) vogliamo rappresentare le rispettive informazioni, i servizi e/o strutture offerti, discipline incluse.

2.2.4. Frasi relative a servizio

Per ogni servizio vogliamo rappresentare il titolo (e.g., corso, lezione una-tantum, prenotazione struttura, consulenza), la descrizione, la durata, il costo, la disciplina, il professionista eventualmente coinvolto, la struttura eventualmente coinvolta.

2.2.5. Frasi relative a invito

Per l'invito è di interesse rappresentare l'utente che l'ha creato, la descrizione dell'evento che si sta organizzando, il servizio coinvolto, gli invitati. Inoltre, si vuole tenere traccia dell'invito accettato o meno, inclusa la data di accettazione o rifiuto.

2.2.6. Frasi relative per acquisto

Per l'acquisto vogliamo rappresentare le informazioni relative al servizio acquistato dagli utenti.

2.2.7. Frasi relative a professionista

Per il professionista vogliamo rappresentare le sue informazioni, l'eventuale società che lo coinvolge, i servizi e/o strutture offerti, discipline incluse.

2.2.8. Frasi relative a struttura

Per la struttura vogliamo rappresentare le varie tipologie (e.g., campi da calcio, da tennis, etc.) offerte dalla società o dal professionista.

2.3. Identificazione delle operazioni principali

Operazione 1: inserisci un nuovo utente indicando tutti i dati (in media 50 volte al giorno)

Operazione 2: inserisci una nuova società indicando i suoi servizi, i suoi dati e i professionisti eventualmente coinvolti (in media 25 volte a settimana)

Operazione 3: inserisci un nuovo acquisto, indicando l'utente che ha acquistato il servizio e la società che lo eroga, indicando eventualmente il professionista coinvolto, la lista degli utenti che partecipano all'evento, la durata del servizio e il suo costo totale (in media 200 volte al giorno)

Operazione 4: per ogni servizio desiderato stampa la società o il professionista che lo offrono al costo minore (in media 200 volte al giorno)

Operazione 5: per ogni società stampa l'elenco dei servizi che offre in merito ad una specifica località indicata dall'utente (in media 200 volte al giorno)

Operazione 6: scegli servizio, località, data e ora, stampa l'elenco dei professionisti disponibili (in media 100 volte al giorno)

Operazione 7: stampa l'elenco dei servizi offerti, dal più richiesto al meno richiesto, identificando il numero degli acquisti per ogni servizio (in media 1 volta al giorno)

3. Progettazione Concettuale

Workpackage	Task	Responsabile
WP1	Progettazione Concettuale	Savarese Marco

3.1. Schema Concettuale

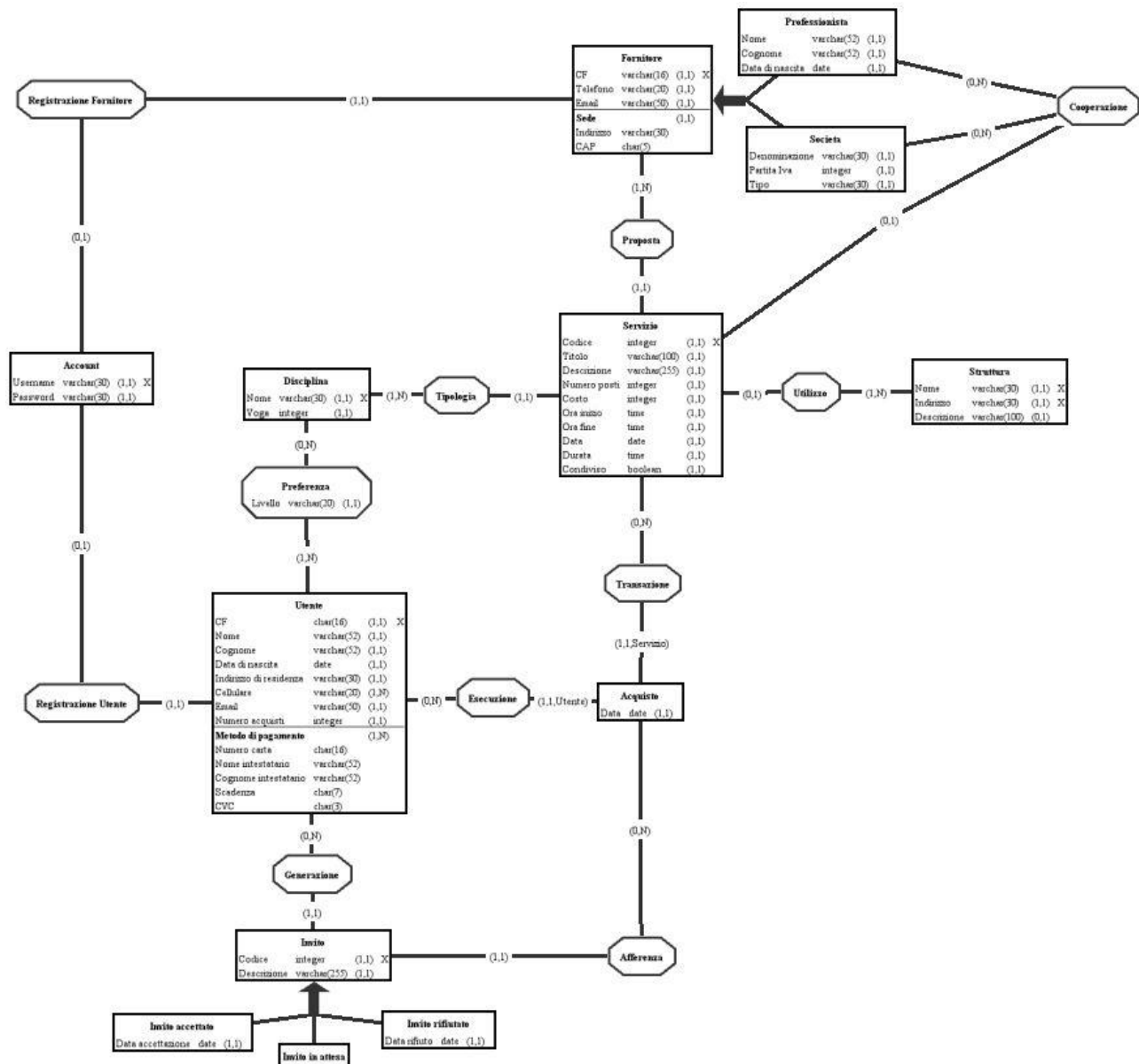


Figura 1. Schema E-R

3.1.1. Note sullo schema E-R

Col supporto del glossario dei termini e delle frasi individuate nel WP0, siamo giunti alla rappresentazione del seguente schema scheletro:

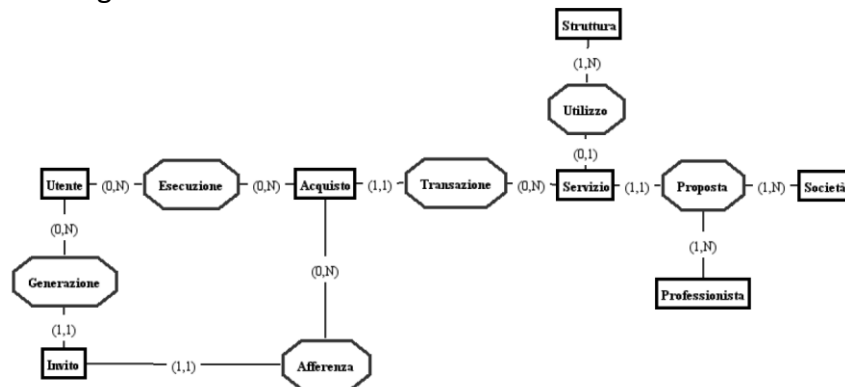


Figura 2. Schema scheletro

La strategia adottata per la descrizione della nostra realtà di interesse è mista. Dopo la definizione dello schema scheletro, siamo andati avanti sia per raffinamenti successivi, sia per estensioni dello schema (come, ad esempio, l'individuazione del concetto di "Account" o la necessità di rappresentare "Disciplina" come entità) per giungere poi alla definizione finale dello stesso.

3.2. Design Pattern

3.2.1. Pattern evoluzione di un concetto

Vogliamo rappresentare il fatto che l'invito subisce un'evoluzione nel tempo, in quanto può essere accettato, rifiutato o in attesa. Pertanto, abbiamo tante generalizzazioni quanti sono gli stati dell'invito.

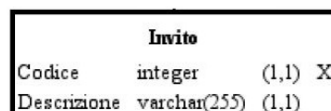


Figura 3. Schema precedente all'applicazione del Pattern Evoluzione di un concetto

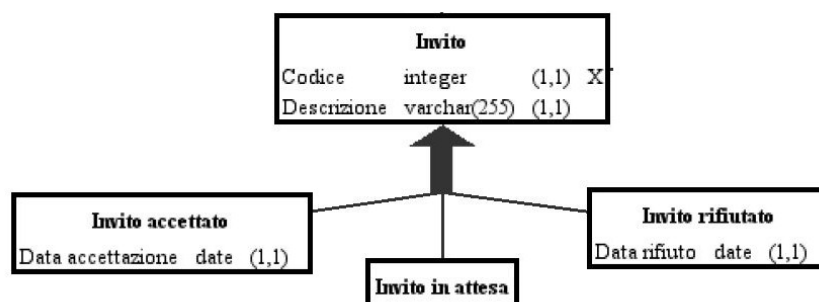


Figura 4. Schema successivo all'applicazione del Pattern Evoluzione di un concetto

3.2.2. Pattern reificazione di un attributo (disciplina)

Nel momento in cui stavamo arricchendo di attributi l'entità "utente", abbiamo introdotto "disciplina". Essa permette all'utente di decidere quale siano le sue discipline preferite e di scegliere, per ognuna di esse, il livello di esperienza. Continuando il processo iterativo di raffinamento, abbiamo notato che "disciplina" era un concetto piuttosto rilevante per la nostra applicazione, ragione per la quale abbiamo deciso di reificare tale concetto, inizialmente concepito come attributo.

Infine, all'entità "disciplina" abbiamo introdotto l'attributo voga.

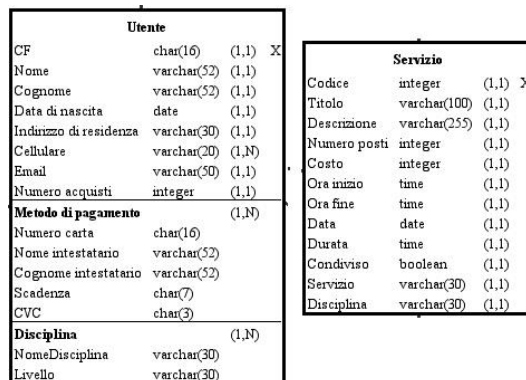


Figura 5. Schema precedente all'applicazione del Pattern reificazione di un attributo (disciplina)

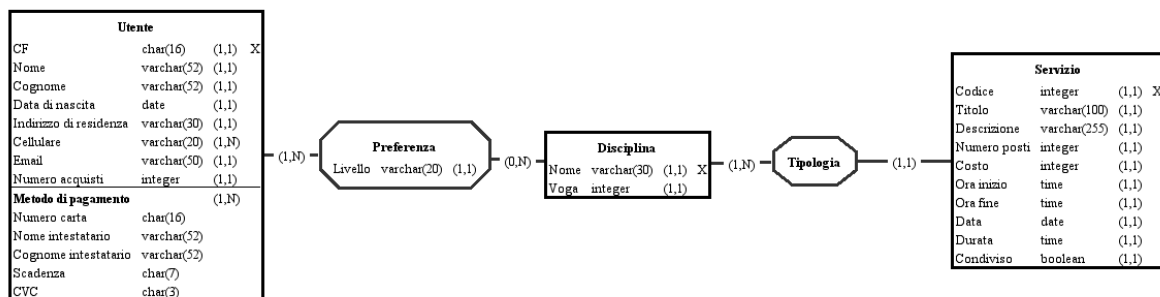


Figura 6. Schema successivo all'applicazione del Pattern reificazione di un attributo (disciplina)

3.2.3. Pattern reificazione di un attributo (account)

Inizialmente, analizzando le frasi relative a “utente”, abbiamo inserito in tale entità l’attributo composto username + password. Successivamente, comprendendo meglio la traccia e considerando l’associazione dello stesso attributo composto anche all’entità “Fornitore”, abbiamo ritenuto necessaria la reificazione dell’attributo.

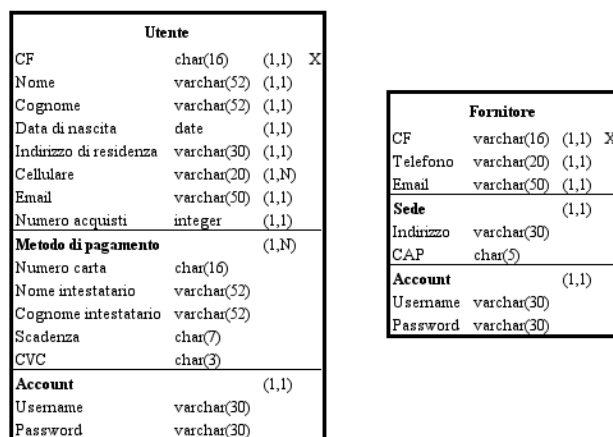


Figura 7. Schema precedente all'applicazione del Pattern reificazione di un attributo (account)

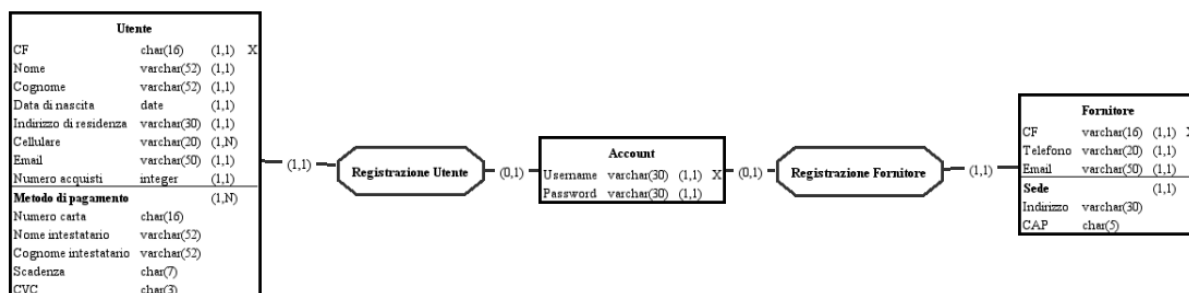


Figura 8. Schema successivo all'applicazione del Pattern reificazione di un attributo (account)

3.3. Dizionario dei Dati

Entità	Descrizione	Attributi	Identificatore
Utente	Persona registrata alla piattaforma che acquista servizi e in alcuni casi genera inviti	CF, Nome, Cognome, Data di nascita, Indirizzo di residenza, Cellulare, Email, Numero acquisti, Metodo di Pagamento (Numero carta, Nome intestatario, Cognome intestatario, Scadenza, CVC)	CF
Acquisto	Acquisto effettuato da un utente	Data	Utente (identificatore esterno) + Servizio (identificatore esterno)
Servizio	Servizio offerto da un fornitore	Codice, Titolo, Descrizione, Numero posti, Costo, Ora inizio, Ora fine, Data, Durata, Condiviso	Codice
Disciplina	Disciplina di un servizio e/o preferita da un utente	Nome, Voga	Nome
Fornitore	Fornitore che offre servizi e/o strutture	CF, Telefono, Email, Sede (Indirizzo, CAP)	CF
Società	Società che offre servizi e/o strutture	Denominazione, Partita Iva, Tipo, CF, Telefono, Email, Sede (Indirizzo, CAP)	CF
Professionista	Persona che offre servizi e/o strutture per mezzo di una società o autonomamente	Nome, Cognome, Data di nascita, CF, Telefono, Email, Sede (Indirizzo, CAP)	CF
Struttura	Luogo fisico in cui si svolge un servizio	Nome, Indirizzo, Descrizione	Nome + Indirizzo
Account	Credenziali di accesso alla piattaforma	Username, Password	Username
Invito	Invito di un utente per altri utenti per la partecipazione ad un servizio	Codice, Descrizione	Codice
Invito accettato	Invito accettato dall'utente invitato	Data accettazione, Codice, Descrizione	Codice

Invito rifiutato	Invito rifiutato dall'utente invitato	Data rifiuto, Codice, Descrizione	Codice
Invito in attesa	Invito che non è stato accettato o rifiutato	Codice, Descrizione	Codice

Tabella 1. Dizionario dei dati – Entità

Relazioni	Descrizione	Entità Coinvolte	Attributi
Generazione	Associa un utente a un invito	Utente (0,N), Invito (1,1)	
Afferenza	Associa un invito a un determinato acquisto	Invito (1,1), Acquisto (0,N)	
Registrazione Utente	Associa un utente a un account	Utente (1,1), Account (0,1)	
Esecuzione	Associa l'utente a un acquisto da egli effettuato	Utente (0,N), Acquisto (1,1)	
Transazione	Associa un acquisto al servizio disponibile acquistato	Acquisto (1,1), Servizio (0,N)	
Preferenza	Associa all'utente la disciplina preferita	Utente (1,N), Disciplina (0,N)	Livello
Tipologia	Associa la disciplina a un servizio	Disciplina (1,N), Servizio (1,1)	
Utilizzo	Associa la struttura ad un servizio	Servizio (0,1), Struttura (1,N)	
Proposta	Associa un servizio a un fornitore	Servizio (1,1), Fornitore (1,N)	
Registrazione Fornitore	Associa un fornitore a un account	Account (0,1), Fornitore (1,1)	
Cooperazione	Associa il professionista con il rispettivo servizio a una società	Professionista (0,N), Società (0,N), Servizio (0,1)	

Tabella 2. Dizionario dei dati - Relazioni

Workpackage	Task	Responsabile
WP4	Regole Aziendali	Paolillo Alessandro

3.4. Regole Aziendali

Regole di Vincolo
<p>(RV1) L'utente per poter invitare un altro utente deve aver acquistato un servizio condiviso e la data dell'invito deve essere precedente alla data di inizio del servizio.</p> <p>(RV2) L'username di un account deve essere univoco all'interno dell'applicazione, per cui utenti e fornitori non devono mai avere lo stesso username.</p> <p>(RV3) Un servizio non deve essere più acquistabile quando la data di inizio diventa uguale o precedente alla data attuale oppure quando il numero di occorrenze di "Transazione" relativamente all'acquisto di quel determinato raggiunge "Numero posti".</p> <p>(RV4) L'ora di fine deve essere successiva all'ora di inizio.</p> <p>(RV5) Ad un professionista che coopera per una società deve essere associato un servizio.</p> <p>(RV6) Un servizio di cooperazione deve essere proposto dalla società</p>

Tabella 3. Regole di vincolo

Regole di derivazione
<p>(RD1) L'ora di fine di un servizio si ottiene sommando la durata all'ora di inizio.</p> <p>(RD2) Il numero di acquisti relativi a ogni singolo utente si ottiene contando le occorrenze di un acquisto.</p> <p>(RD3) La disciplina più in "Voga" si ottiene contando le occorrenze in transazione relative a quella particolare disciplina.</p>

Tabella 4. Regole di derivazione

4. Progettazione Logica

Workpackage	Task	Responsabile
WP2	Progettazione Logica	Landi Thomas

4.1. Ristrutturazione Schema Concettuale

4.1.1. Analisi delle Prestazioni

4.1.1.1. Tavola dei volumi

Concetto	Tipo	Volume
Fornitore	E	6800
Professionista	E	5600
Società	E	1200
Servizio	E	8000
Struttura	E	4400
Disciplina	E	100
Utente	E	18000
Account	E	24800
Acquisto	E	73000
Invito	E	65000
Invito accettato	E	54750
Invito rifiutato	E	10000
Invito in attesa	E	250
Cooperazione	R	3600
Registrazione fornitore	R	6800
Registrazione utente	R	18000
Proposta	R	8000
Tipologia	R	8000
Utilizzo	R	7000
Preferenza	R	54000
Transazione	R	73000
Esecuzione	R	73000
Generazione	R	65000
Afferenza	R	65000

Tabella 5. Tavola dei volumi

4.1.1.2. Tavola delle operazioni

Operazione	Tipo	Frequenza
Operazione 1: inserisci utente	I	50/giorno
Operazione 2: inserisci società	I	25/settimana
Operazione 3: inserisci acquisto	I	200/giorno
Operazione 4: stampa società più economica per il servizio scelto	I	200/giorno
Operazione 5: stampa elenco servizi	I	200/giorno
Operazione 6: stampa professionisti per un determinato servizio	I	100/giorno
Operazione 7: stampa servizi dal più richiesto	B	1/giorno
Operazione 8: stampa le informazioni su una disciplina (incluso il numero di volte in cui è stata acquistata)	B	1/settimana
Operazione 9: stampa il numero di acquisti di un utente	B	1/settimana

Tabella 6. Tavola delle operazioni

4.2. Analisi delle ridondanze

- Ridondanza 1: Numero acquisti**
 L'attributo "Numero acquisti" nell'entità "Utente" è ridondante in quanto è possibile tener traccia di tale informazione contando il numero di occorrenze presenti in "Esecuzione".
 Il vincolo di derivazione coinvolto è il RD2, del tipo "Attributo Derivabile da conteggio di occorrenze".
- Ridondanza 2: Voga**
 L'attributo "Voga" nell'entità "Disciplina" è ridondante in quanto è possibile tener traccia di tale informazione contando il numero di occorrenze presenti in "Transazione".
 Il vincolo di derivazione coinvolto è il RD3, del tipo "Attributo Derivabile da conteggio di occorrenze".
- Ridondanza 3: Ora fine**
 L'attributo "Ora fine" nell'entità "Servizio" è ridondante in quanto è possibile ricavare tale informazione dagli attributi "Ora inizio" e "Durata".
 Il vincolo di derivazione coinvolto è il RD1, del tipo "Attributo Derivabile nella stessa entità": proprio per questo motivo l'analisi della ridondanza risulterebbe abbastanza banale. Inserendo un nuovo servizio, infatti, tale attributo non incide sul numero di accessi, ma solo sulla memoria con una quantità alquanto irrilevante.

4.2.1. Analisi della ridondanza 1: Numero acquisti

- **Operazione 3: inserire acquisto**

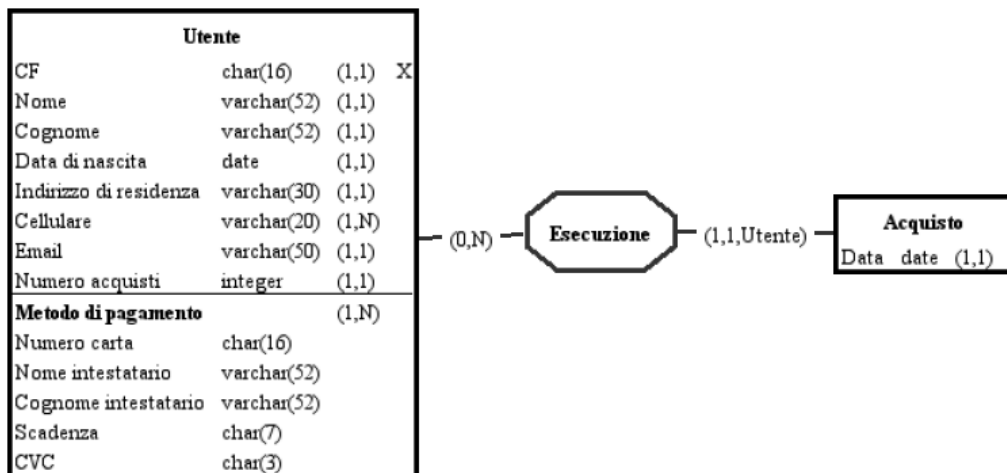


Figura 9. Percorso dell'operazione per l'analisi della ridondanza "Numero acquisti"

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
ACQUISTO	E	1	S
ESECUZIONE	R	1	S
UTENTE	E	1	L
UTENTE	E	1	S

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
ACQUISTO	E	1	S
ESECUZIONE	R	1	S

- **Operazione 9: stampa il numero di acquisti di un utente**

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
UTENTE	E	1	L

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
UTENTE	E	1	L
ESECUZIONE	R	4	L

4.2.1.1. Valutazione della ridondanza 1: Numero Acquisti

Dopo aver analizzato le operazioni che coinvolgono la ridondanza si osserva che, con il carico considerato:

- In presenza di ridondanza il costo delle operazioni è di circa 1400 accessi giornalieri (dovuta alla sola analisi dell'operazione 3, in quanto il costo dell'operazione 9 è trascurabile)
- L'occupazione di memoria è di circa 72000 byte
- In assenza di ridondanza il costo delle operazioni è di 800 accessi giornalieri (dovuta alla sola analisi dell'operazione 3, in quanto il costo dell'operazione 9 è trascurabile)

Pertanto, si decide di non mantenere la ridondanza in quanto aumenta il numero di accessi.

4.2.2. Analisi della ridondanza 2: Voga

- **Operazione 3: Operazione aggiungi acquisto**

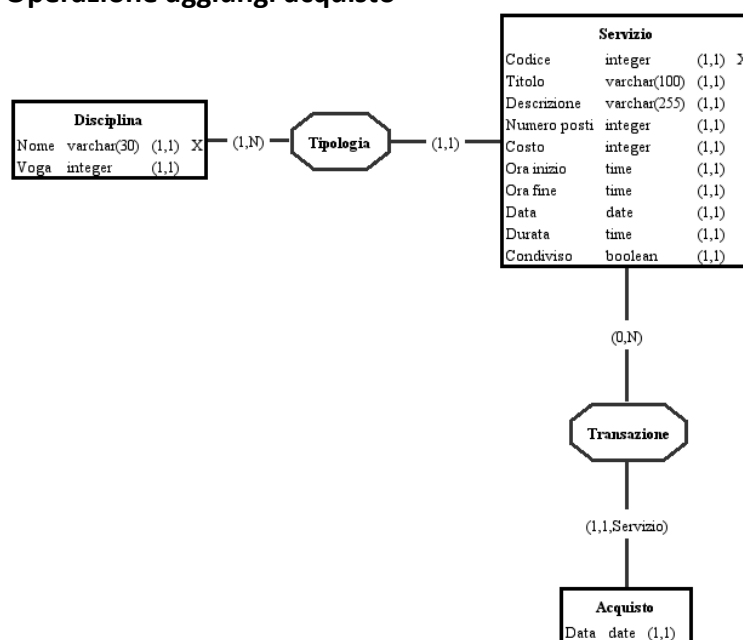


Figura 10. Percorso dell'operazione per l'analisi della ridondanza "Voga"

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
ACQUISTO	E	1	S
TRANSAZIONE	R	1	S
SERVIZIO	E	1	L
TIPOLOGIA	R	1	L
DISCIPLINA	E	1	L
DISCIPLINA	E	1	S

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
ACQUISTO	E	1	S
TRANSAZIONE	R	1	S

- **Operazione 8: stampa le informazioni su una disciplina (incluso il numero di volte in cui è stata acquistata)**

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
DISCIPLINA	E	1	L

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
DISCIPLINA	E	1	L
TIPOLOGIA	R	80	L
SERVIZIO	E	80	L
TRANSAZIONE	R	720	L

4.2.2.1. Valutazione della ridondanza 2: Voga

Dopo aver analizzato le operazioni che coinvolgono la ridondanza si osserva che, con il carico considerato:

- In presenza di ridondanza il costo delle operazioni è di circa 1800 accessi giornalieri (dovuta alla sola analisi dell'operazione 3, in quanto il costo dell'operazione 8 è trascurabile)
- L'occupazione di memoria è di circa 400 byte
- In assenza di ridondanza il costo delle operazioni è di 925 accessi giornalieri (800 per l'operazione 3 e circa 125 per l'operazione 8)

Pertanto, si decide di non mantenere la ridondanza in quanto aumenta il numero di accessi.

4.3. Eliminazione delle generalizzazioni

4.3.1. Generalizzazione *Fornitore*

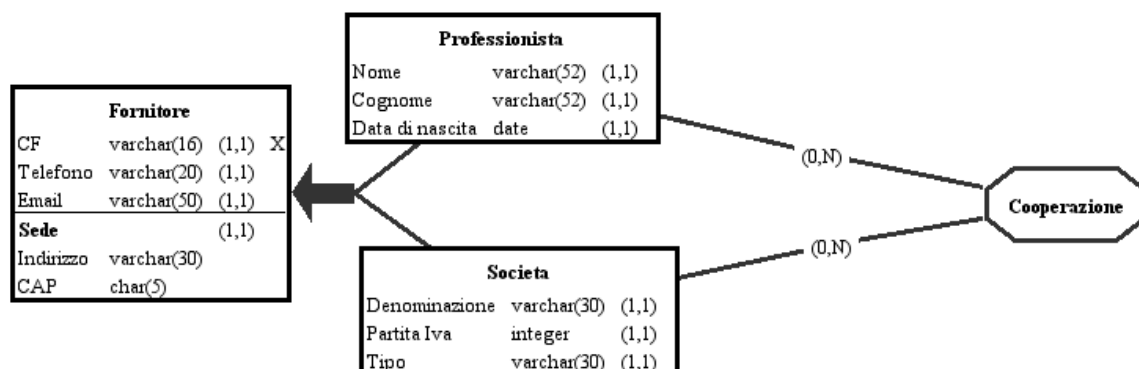


Figura 11. Porzione di schema che coinvolge la generalizzazione "Fornitore"

Dallo schema concettuale osserviamo che Fornitore (padre), Società (figlia) e Professionista (figlia) realizzano insieme una generalizzazione totale. Abbiamo ritenuto opportuno l'accorpamento del genitore della generalizzazione nelle figlie.

Questa soluzione è conveniente quando ci sono operazioni che si riferiscono solo a occorrenze di Società e di Professionisti e dunque fanno distinzioni tali entità. Pur duplicando gli attributi, ottengo comunque un risparmio di memoria perché non avremo mai valori nulli.

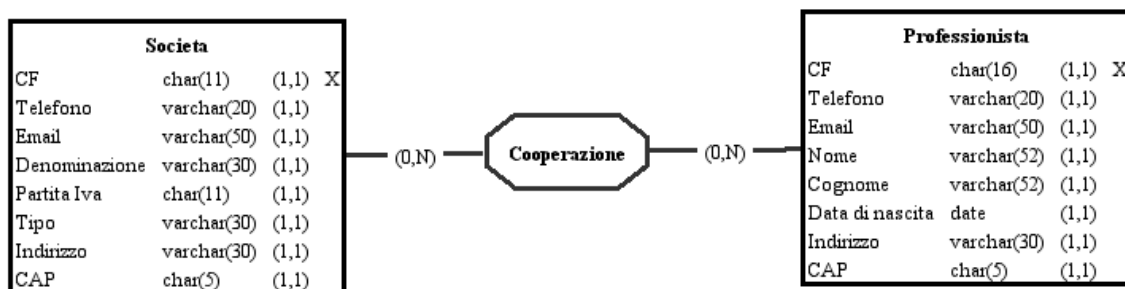


Figura 12. Porzione di schema modificata che coinvolge la generalizzazione "Fornitore"

4.3.2. Generalizzazione *Invito*

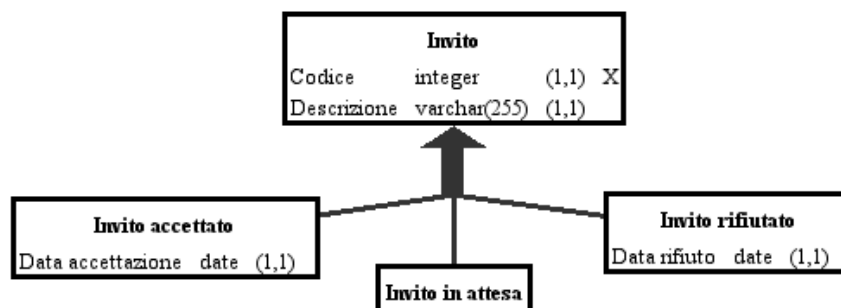


Figura 13. Porzione di schema che coinvolge la generalizzazione "Invito"

In questo caso la strada scelta è stata l'accorpamento delle figlie della generalizzazione nel genitore: le tre entità figlie vengono eliminate e le loro proprietà vengono aggiunte all'entità genitore.

All'entità genitore è stato aggiunto un ulteriore attributo ("Esito") che serve a distinguere l'esito di un'occorrenza del padre (cioè se apparteneva a Invito accettato, Invito in attesa o Invito rifiutato). Inoltre gli attributi "Data accettazione" e "Data rifiuto", una volta accorpati nell'entità padre, sono stati "fusi" in "Data esito" perché concettualmente avrebbero rappresentato la stessa cosa e per evitare che uno dei due fosse sempre nullo.

La scelta assicura un numero minore di accessi tra entità.

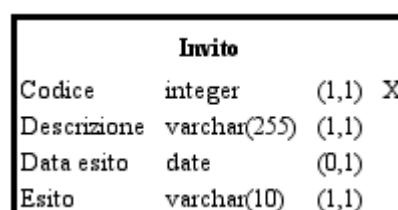


Figura 14. Porzione di schema modificata che coinvolge la generalizzazione "Fornitore"

4.4. Partizionamento/Accorpamento Entità e Associazioni

Non abbiamo ritenuto necessario nessun Partizionamento/Accorpamento Entità e Associazioni. L'unica modifica necessaria in questa fase è stata l'eliminazione degli unici due attributi multivalore presenti nello schema concettuale: Cellulare e Metodo di pagamento, entrambi presenti nell'entità "Utente". Questa ristrutturazione si rende necessaria perché, come per le generalizzazioni, il modello relazionale non permette di rappresentare in maniera diretta questo tipo di attributo.

Utente			
CF	char(16)	(1,1)	X
Nome	varchar(52)	(1,1)	
Cognome	varchar(52)	(1,1)	
Data di nascita	date	(1,1)	
Indirizzo di residenza	varchar(30)	(1,1)	
Cellulare	varchar(20)	(1,N)	
Email	varchar(50)	(1,1)	
Numero acquisti	integer	(1,1)	
Metodo di pagamento			
Numero carta	char(16)		
Nome intestatario	varchar(52)		
Cognome intestatario	varchar(52)		
Scadenza	char(7)		
CVC	char(3)		

Figura 15. Porzione di schema che attributi multivalore

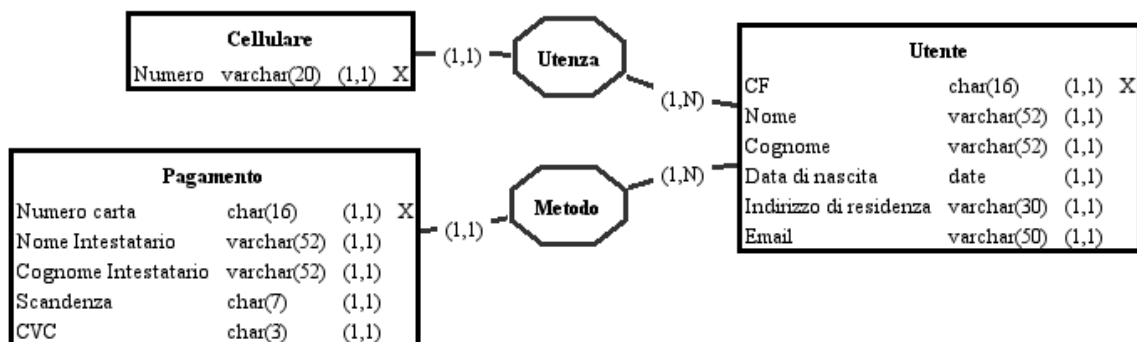


Figura 16. Porzione di schema ristrutturata dopo l'eliminazione di attributi multivalore

4.5. Scelta degli identificatori principali

Un identificatore interno è sempre preferibile a uno esterno, per questo motivo in “Acquisto” abbiamo introdotto un “ID” da utilizzare come identificatore.

Anche per le entità “Struttura” e “Disciplina” abbiamo inserito un “ID”; “Servizio” e “Invito”, che avevano già il proprio codice, continuano a mantenere lo stesso identificatore.

Per quanto riguarda le altre entità (Account, Società, Professionista, Utente, Cellulare, Pagamento) abbiamo ritenuto che rispettivamente i CF, il Numero di cellulare, il Numero di carta da utilizzare e l'Username rispecchino le caratteristiche richieste.

4.6. Schema ristrutturato finale

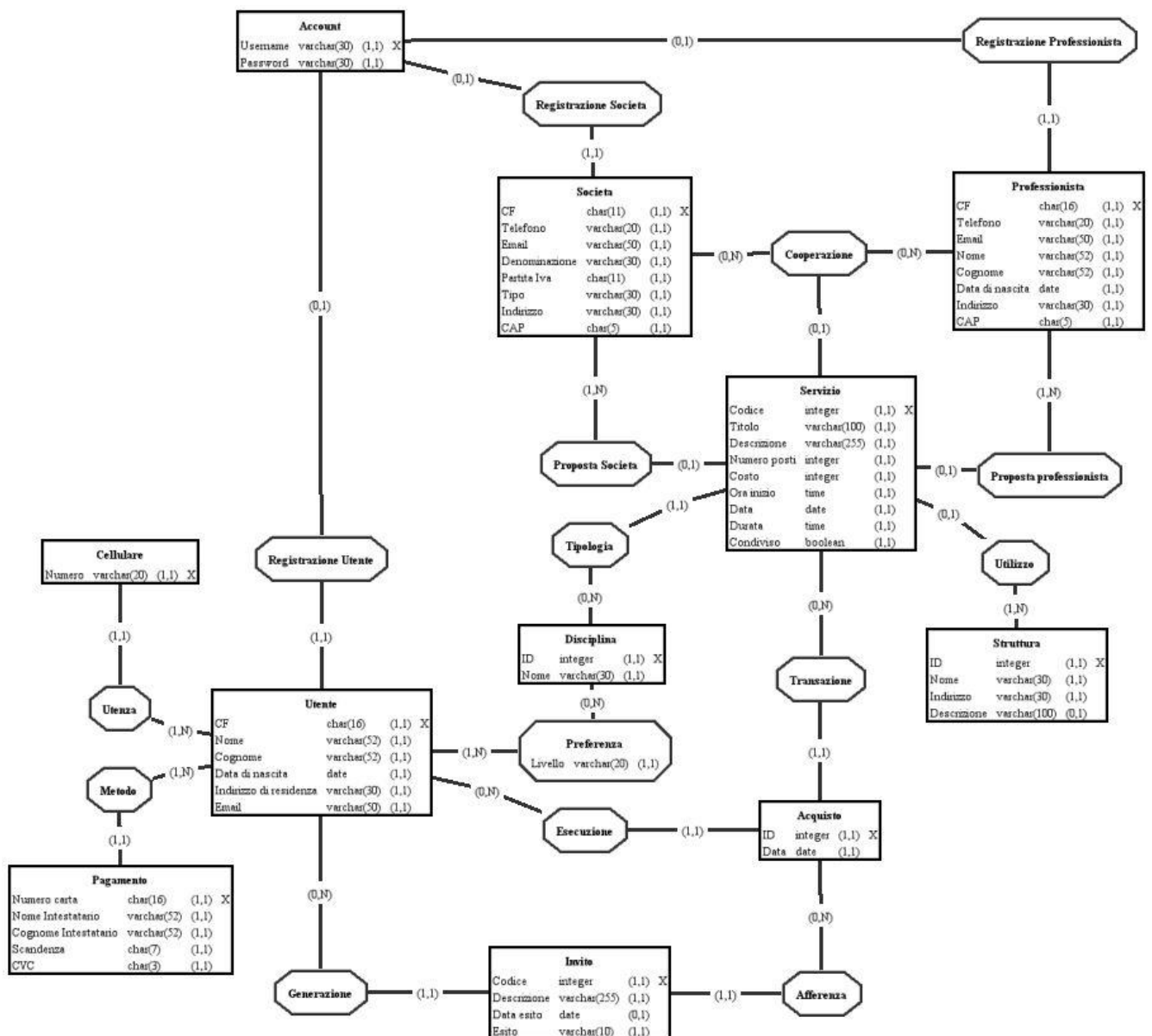
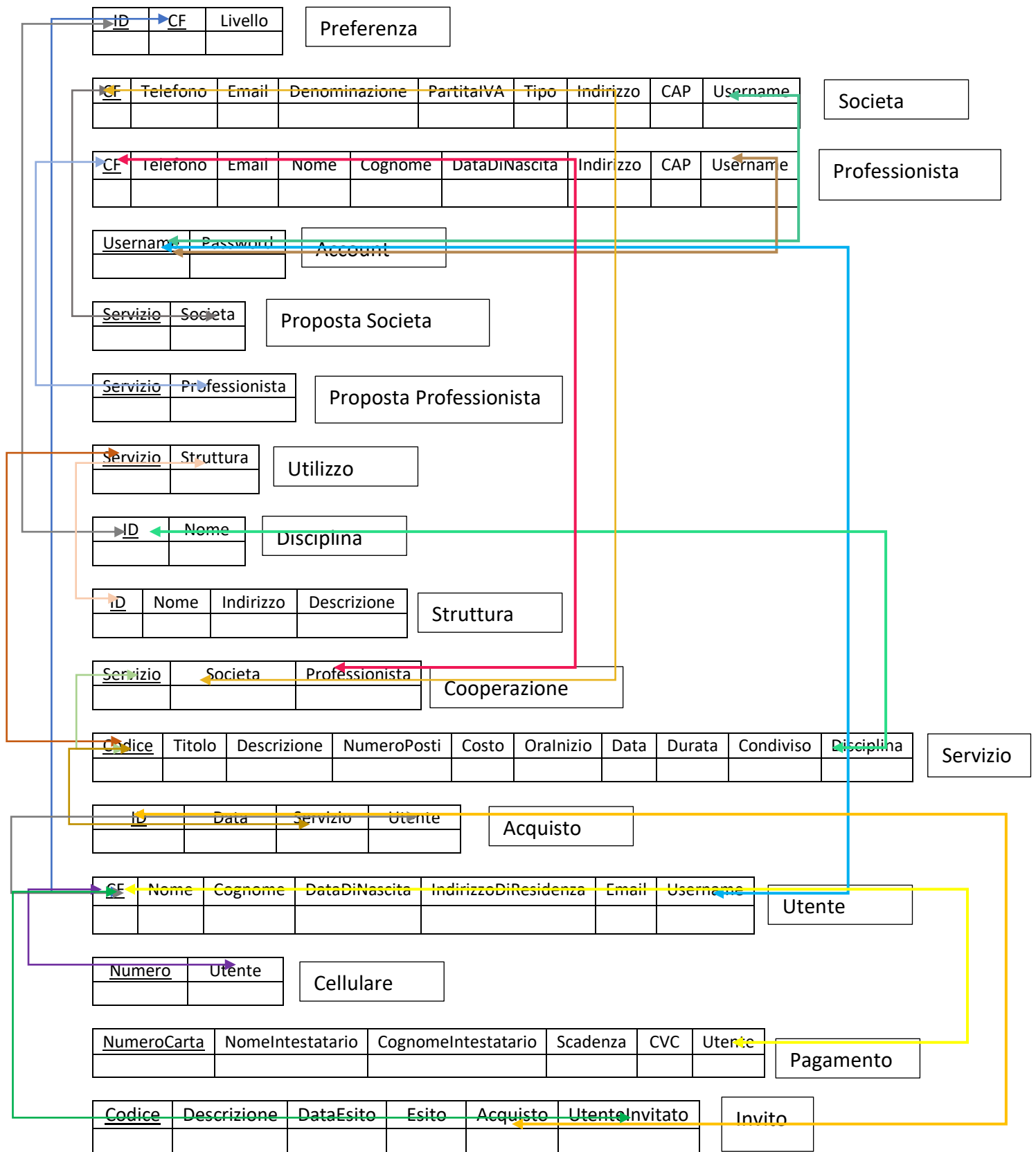


Figura 17. Schema ER Ristrutturato

4.7. Schema logico

- PREFERENZA (ID, CF, Livello)
- SOCIETA (CF, Telefono, Email, Denominazione, PartitaIVA, Tipo, Indirizzo, CAP, Username)
- PROFESSIONISTA (CF, Telefono, Email, Nome, Cognome, DataDiNascita, Indirizzo, CAP, Username)
- ACCOUNT (Username, Password)
- PROPOSTA SOCIETA (Servizio, Societa)
- PROPOSTA PROFESSIONISTA (Servizio, Professionista)
- UTILIZZO (Servizio, Struttura)
- DISCIPLINA (ID, Nome)
- STRUTTURA (ID, Nome, Indirizzo, Descrizione)
- COOPERAZIONE (Servizio, Società, Professionista)
- SERVIZIO (Codice, Titolo, Descrizione, NumeroPosti, Costo, OraInizio, Data, Durata, Condiviso, Disciplina)
- ACQUISTO (ID, Data, Servizio, Utente)
- UTENTE (CF, Nome, Cognome, DataDiNascita, IndirizzoDiResidenza, Email, Username)
- CELLULARE (Numero, Utente)
- PAGAMENTO (NumeroCarta, NomeIntestatario, CognomeIntestatario, Scadenza, CVC, Utente)
- INVITO (Codice, Descrizione, DataEsito, Esito, Acquisto, UtenteInvitato)

4.8. Documentazione dello schema logico



5. Normalizzazione

Workpackage	Task	Responsabile
WP3	Normalizzazione	Memoli Matteo

5.1. Analisi delle dipendenze funzionali

5.1.1. Preferenza

PREFERENZA (ID, CF, Livello)

ID, CF -> Livello

Dall'analisi delle dipendenze funzionali possiamo affermare che la relazione

Preferenza soddisfa la BCNF

5.1.2. Societa

SOCIETA (CF, Telefono, Email, Denominazione, PartitaIVA, Tipo, Indirizzo, CAP, Username)

CF -> Telefono, Email, Denominazione, PartitaIVA, Tipo, Indirizzo, CAP, Username

Dall'analisi delle dipendenze funzionali possiamo affermare che la relazione **Societa** soddisfa la BCNF

5.1.3. Professionista

PROFESSIONISTA (CF, Telefono, Email, Nome, Cognome, DataDiNascita, Indirizzo, CAP, Username)

CF -> Telefono, Email, Nome, Cognome, DataDiNascita, Indirizzo, CAP, Username

Dall'analisi delle dipendenze funzionali possiamo affermare che la relazione

Professionista soddisfa la BCNF

5.1.4. Account

ACCOUNT (Username, Password)

Username -> Password

Dall'analisi delle dipendenze funzionali possiamo affermare che la relazione **Account** soddisfa la BCNF

5.1.5. Proposta Societa

PROPOSTA SOCIETA (Servizio, Societa)

Servizio -> Societa

Dall'analisi delle dipendenze funzionali possiamo affermare che la relazione

Proposta Societa soddisfa la BCNF

5.1.6. Proposta professionista

PROPOSTA PROFESSIONISTA (Servizio, Professionista)

Servizio -> Professionista

Dall'analisi delle dipendenze funzionali possiamo affermare che la relazione

Proposta Professionista soddisfa la BCNF

5.1.7. Utilizzo

UTILIZZO (Servizio, Struttura)

Servizio -> Struttura

Dall'analisi delle dipendenze funzionali possiamo affermare che la relazione **Utilizzo** soddisfa la BCNF

5.1.8. Disciplina

DISCIPLINA (ID, Nome)

ID -> Nome

Dall'analisi delle dipendenze funzionali possiamo affermare che la relazione

Disciplina soddisfa la BCNF

5.1.9. Struttura

STRUTTURA (ID, Nome, Indirizzo, Descrizione)

ID -> Nome, Indirizzo, Descrizione

Dall'analisi delle dipendenze funzionali possiamo affermare che la relazione

Struttura soddisfa la BCNF

5.1.10. Cooperazione

COOPERAZIONE (Servizio, Societa, Professionista)

Servizio -> Societa, Professionista

Dall'analisi delle dipendenze funzionali possiamo affermare che la relazione

Cooperazione soddisfa la BCNF

5.1.11. Servizio

SERVIZIO (Codice, Titolo, Descrizione, NumeroPosti, Costo, Orainizio, Data, Durata, Condiviso, Disciplina)

Codice -> Titolo, Descrizione, NumeroPosti, Costo, Orainizio, Data, Durata, Condiviso, Disciplina

Dall'analisi delle dipendenze funzionali possiamo affermare che la relazione **Servizio** soddisfa la BCNF

5.1.12. Acquisto

ACQUISTO (ID, Data, Servizio, Utente)

ID -> Data, Servizio, Utente

Dall'analisi delle dipendenze funzionali possiamo affermare che la relazione

Acquisto soddisfa la BCNF

5.1.13. Utente

UTENTE (CF, Nome, Cognome, DataDiNascita, IndirizzoDiResidenza, Email, Username)

CF -> Nome, Cognome, DataDiNascita, IndirizzoDiResidenza, Email, Username

Dall'analisi delle dipendenze funzionali possiamo affermare che la relazione **Utente** soddisfa la BCNF

5.1.14. Cellulare

CELLULARE (Numero, Utente)

Numero -> Utente

Dall'analisi delle dipendenze funzionali possiamo affermare che la relazione

Cellulare soddisfa la BCNF

5.1.15. Pagamento

PAGAMENTO (NumeroCarta, NomeIntestatario, CognomeIntestatario, Scadenza, CVC, Utente)

NumeroCarta -> NomeIntestatario, CognomeIntestatario, Scadenza, CVC, Utente

Dall'analisi delle dipendenze funzionali possiamo affermare che la relazione

Pagamento soddisfa la BCNF

5.1.16. Invito

INVITO (Codice, Descrizione, DataEsito, Esito, Acquisto, UtenteInvitato)

Codice -> Descrizione, DataEsito, Esito, Acquisto, UtenteInvitato

Dall'analisi delle dipendenze funzionali possiamo affermare che la relazione **Invito** soddisfa la BCNF

6. Script Creazione e Popolamento Database

Workpackage	Task	Responsabile
WP2	SQL: Script creazione e popolamento	Landi Thomas

6.1. Script di Creazione del Database

6.1.1. Script di Drop

```

DROP TABLE IF EXISTS Preferenza cascade;
DROP TABLE IF EXISTS Societa cascade;
DROP TABLE IF EXISTS Professionista cascade;
DROP TABLE IF EXISTS Account cascade;
DROP TABLE IF EXISTS PropostaSocieta cascade;
DROP TABLE IF EXISTS PropostaProfessionista cascade;
DROP TABLE IF EXISTS Utilizzo cascade;
DROP TABLE IF EXISTS Disciplina cascade;
DROP TABLE IF EXISTS Struttura cascade;
DROP TABLE IF EXISTS Cooperazione cascade;
DROP TABLE IF EXISTS Servizio cascade;
DROP TABLE IF EXISTS Acquisto cascade;
DROP TABLE IF EXISTS Utente cascade;
DROP TABLE IF EXISTS Cellulare cascade;
DROP TABLE IF EXISTS Pagamento cascade;
DROP TABLE IF EXISTS Invito cascade;
DROP DOMAIN IF EXISTS TipoEsito cascade;
DROP DOMAIN IF EXISTS TipoLivello cascade;

```

6.1.2. Creazione Domini

```

create domain TipoEsito as varchar(10)
check (value = 'Accettato' or value = 'Rifiutato' or value = 'In attesa');

create domain TipoLivello as varchar(20)
check(value = 'Amatoriale' or value = 'Novizio' or value = 'Agonistico');

```

6.1.3. Creazione Tabelle

```

create table Disciplina (
    ID serial primary key,
    Nome varchar(30) not NULL UNIQUE
);

create table Account(
    Username varchar(30)Primary Key,
    Password varchar(30)not NULL
);

```

```
create table Societa(  
    CF char(11) Primary Key,  
    Telefono varchar(20) not NULL,  
    Email varchar(50) not NULL,  
    Denominazione varchar(30),  
    Partitalva char (11) not NULL UNIQUE,  
    Tipo varchar(30),  
    Indirizzo varchar(30),  
    CAP char(5) default '00000',  
    Username varchar(30) UNIQUE,  
    constraint Foreign_Societa Foreign Key (Username) references Account on update restrict  
on delete set NULL  
);
```

```
create table Professionista(  
    CF char(16) Primary Key,  
    Telefono varchar(20) not NULL,  
    Email varchar(50) not NULL,  
    Nome varchar(52) not NULL,  
    Cognome varchar(52) not NULL,  
    DataDiNascita date,  
    Indirizzo varchar(30),  
    CAP char(5) default '00000',  
    Username varchar(30) UNIQUE,  
    constraint Foreign_Professionista Foreign Key (Username) references Account on update  
restrict on delete set NULL  
);
```

```
create table Utente(  
    CF char(16) Primary key,  
    Nome varchar(52) not NULL,  
    Cognome varchar(52) not NULL,  
    DataDiNascita date not NULL,  
    IndirizzoDiResidenza varchar(30) not NULL,  
    Email varchar(50) not NULL,  
    Username varchar(30) UNIQUE,  
    constraint Foreign_Utente foreign key(Username) references Account on update restrict on  
delete set NULL  
);
```

```
create table Cellulare(  
    Numero varchar(20) Primary key,  
    Utente varchar(16) not NULL,  
    constraint Foreign_Cellulare foreign key (Utente) references Utente on update cascade on  
delete restrict  
);
```

```
create table Pagamento(  
    NumeroCarta char(16) Primary key,  
    NomeIntestatario varchar(52) not NULL,  
    CognomeIntestatario varchar(52) not NULL,  
    Scadenza char(7) not NULL,  
    CVC char(3) not null,  
    Utente char(16) not null,  
    constraint Foreign_Pagamento foreign key(Utente) references Utente on update cascade on  
delete restrict,  
    constraint Check_Scadenza Check (Scadenza like '__/20__')  
);
```

```
create table Servizio(  
    Codice serial primary key,  
    Titolo varchar(100) not NULL,  
    Descrizione varchar(255) not NULL,  
    NumeroPosti integer not NULL,  
    Costo decimal(6,2) not NULL,  
    Data date not NULL,  
    OraInizio time not NULL,  
    Durata time not NULL,  
    Condiviso boolean not NULL,  
    Disciplina integer not NULL,  
    constraint Key_Disciplina foreign key (Disciplina) references Disciplina on update restrict on  
delete restrict,  
    constraint CostoMaggioreDiZero CHECK (Costo > 0),  
    constraint NumeroPostiMaggioreDiZero CHECK(NumeroPosti > 0)  
);
```

```
create table PropostaSocieta (  
    Servizio serial primary key,  
    Societa char(11) not NULL,  
    constraint Foreign_PropostaSocieta foreign key (Societa) references Societa on update  
cascade on delete restrict,  
    constraint Foreign_PropostaSocieta_Servizio foreign key (Servizio) references Servizio on  
update restrict on delete restrict  
);
```

```
create table PropostaProfessionista (  
    Servizio serial primary key,  
    Professionista char(16) not NULL,  
    constraint Foreign_PropostaProfessionista foreign key (Professionista) references  
Professionista on update restrict on delete restrict,  
    constraint Foreign_PropostaProposta_Servizio foreign key (Servizio) references Servizio on  
update restrict on delete restrict  
);
```

```
create table Struttura(  
    ID serial primary key,  
    Nome varchar(30) not NULL,  
    Indirizzo varchar(30) not NULL,  
    Descrizione varchar(100)  
);  
  
create table Utilizzo (  
    Servizio serial primary key,  
    Struttura integer not NULL,  
    constraint Foreign_Struttura foreign key (Struttura) references Struttura on update cascade  
on delete restrict,  
    constraint Foreign_StrutturaServizio foreign key (Servizio) references Servizio on update  
restrict on delete restrict  
);  
  
create table Cooperazione(  
    Servizio serial primary key,  
    Societa char(11) not NULL,  
    Professionista char(16) not NULL,  
    constraint Foreign_CooperazioneSocieta foreign key (Societa) references Societa on update  
cascade on delete restrict,  
    constraint Foreign_CooperazioneProfessionista foreign key (Professionista) references  
Professionista on update cascade on delete restrict,  
    constraint Foreign_CooperazioneServizio foreign key (Servizio) references Servizio on  
update restrict on delete restrict  
);  
  
create table Acquisto(  
    ID serial primary key,  
    Data date not NULL,  
    Servizio integer not NULL,  
    Utente char(16) not NULL,  
    constraint Key_Servizio foreign key (Servizio) references Servizio on update restrict on delete  
restrict,  
    constraint Key_Utente foreign key (Utente) references Utente on update cascade on delete  
restrict  
);
```

```
create table Invito(  
    Codice serial Primary key,  
    Descrizione varchar(255) not null,  
    DataEsito date,  
    Esito TipoEsito default 'In attesa',  
    Acquisto integer not null,  
    UtenteInvitato char(16) not null,  
    constraint Foreign_InvitoAcquisto foreign key(Acquisto) references Acquisto on update  
restrict on delete restrict,  
    constraint Foreign_InvitoUtente foreign key(UtenteInvitato) references Utente on update  
cascade on delete restrict  
);
```

```
create table Preferenza(  
    ID serial,  
    CF char(16),  
    Livello TipoLivello not NULL,  
    constraint Key_Preferenza Primary Key (ID,CF),  
    constraint Foreign_Preferenza_Disciplina Foreign Key (ID) references Disciplina on update  
restrict on delete restrict,  
    constraint Foreign_Preferenza_Utente Foreign Key (CF) references Utente on update  
cascade on delete cascade  
);
```

6.2. Script di Popolamento del Database

```
Insert into Disciplina (Nome) values ('Calcio a 8');  
Insert into Disciplina (Nome) values ('Tennis');  
Insert into Disciplina (Nome) values ('Pallavolo' );  
Insert into Disciplina (Nome) values ('Golf');  
Insert into Disciplina (Nome) values ('Basket');  
Insert into Disciplina (Nome) values ('Danza classica');  
Insert into Disciplina (Nome) values ('Hip-Hop');  
Insert into Disciplina (Nome) values ('Calcio a 11');  
Insert into Disciplina (Nome) values ('Calcio a 5');  
Insert into Disciplina (Nome) values ('Ciclismo');  
Insert into Disciplina (Nome) values ('Spinning');  
Insert into Disciplina (Nome) values ('Zumba');  
Insert into Disciplina (Nome) values ('Pallanuoto');  
Insert into Disciplina (Nome) values ('Nuoto');  
Insert into Disciplina (Nome) values ('Canottaggio');  
Insert into Disciplina (Nome) values ('Bocce');  
Insert into Disciplina (Nome) values ('Arti marziali');  
Insert into Disciplina (Nome) values ('Bowling');  
Insert into Disciplina (Nome) values ('Scherma');  
Insert into Disciplina (Nome) values ('Boxe');  
Insert into Disciplina (Nome) values ('Pugilato');  
Insert into Disciplina (Nome) values ('Soft Tennis');
```

Insert into Disciplina (Nome) values ('Ginnastica Artistica');
Insert into Disciplina (Nome) values ('Ski archery');
Insert into Disciplina (Nome) values ('Kung Fu Vietnamita');
Insert into Disciplina (Nome) values ('Freccette');
Insert into Disciplina (Nome) values ('Ginnastica Ritmica');
Insert into Disciplina (Nome) values ('Atletica leggera');
Insert into Disciplina (Nome) values ('Hockey su pista');
Insert into Disciplina (Nome) values ('Sollevamento pesi');

Insert into Account values ('cagliarimarioDG','cmDGgcom');
Insert into Account values ('davincipippoB','dpBgcom');
Insert into Account values ('leopardimariaS','lmSgcom');
Insert into Account values ('dantefrancesca','dfMgcom');
Insert into Account values ('toscanamarcoS','tmSgcom');
Insert into Account values ('romaspa','rspagcom');
Insert into Account values ('napolisnc','nsncgcom');
Insert into Account values ('firenzespas','fspagcom');
Insert into Account values ('milanospa','mspagcom');
Insert into Account values ('torinoss','tssgcom');
Insert into Account values ('GigiEsp89','ge89fcom');
Insert into Account values ('Sara00','ls99gcom');
Insert into Account values ('Paolillo99','ap99gcom');
Insert into Account values ('tlandi3','lt1999gcom');
Insert into Account values ('SavareseGranata19','sm1919gcom');
Insert into Account values ('ValeriaC01','vc01gcom');
Insert into Account values ('Fortunato77','ff77gcom');
Insert into Account values ('Memoli85','mm85gcom');
Insert into Account values ('Serena94','sc94gcom');
Insert into Account values ('Santo90','sg90gcom');

Insert into Societa values
('12345678912','089302230','romaspa@gmail.com','FederazioneRomaSPA','12345678912','Federa
zione','Via Roma','84100','romaspa');
Insert into Societa values
('43215678912','089336712','napolisnc@gmail.com','PalestraNapoliSNC','43215678912','Palestra',
'Via Napoli','94105','napolisnc');
Insert into Societa values
('12341234912','089312019','firenzespas@gmail.com','FederazuineFirenzeSPA.','14321567891','Fe
derazione','Via Firenze','91340','firenzespas');
Insert into Societa values
('12344321912','089282015','milanospa@gmail.com','FederazioneMllanoSPA.','12344321912','Fed
erazione','Via Milano','90100','milanospa');
Insert into Societa values
('12301010912','089102015','torinoss@gmail.com','ClubTorinoSS.','98745678912','Club','Via
Torino','89150','torinoss');

Insert into Professionista values

('DGRMRA99L01L219J','3332534556','cagliarimarioDG@gmail.com','Mario','De Gregori','1999-07-01','Via Cagliari','86700','cagliarimarioDG');

Insert into Professionista values

('BDAPPP86D23H703F','3333644234','davincipippoB@gmail.com','Pippo','Baudo','1986-04-23','Via Da Vinci','95110','davincipippoB');

Insert into Professionista values

('SNTMRA92T59B180C','3334754133','leopardimariaS@gmail.com','Maria','Santos','1992-12-19','Via Leopardi','10101','leopardimariaS');

Insert into Professionista values

('MRRFNC82B45C351S','3335864011','dantefrancescaM@gmail.com','Francesca','Morriello','1982-02-05','Via Dante','90892','dantefrancesca');

Insert into Professionista values

('SBBMRC92M09H703Y','3336974412','toscanamarcoS@gmail.com','Marco','Sabbarese','1992-08-09','Via Toscana','30129','toscanamarcoS');

insert into Utente values ('SPSGNR89T12G942K','Gennaro','Esposito','12/12/1989','Via Nicola Farina','GennaroEsposito89@gmail.com','GigiEsp89');

insert into Utente values ('LNDRA00R41L781V','Sara','Landi','01/10/2000','Corso Emanuele','LandiSara99@gmail.com','Sara00');

insert into Utente values ('PLLSN99C20H703T','Alessandro','Paolillo','20/03/1999','Via Sandro Pertini','APaolillo@gmail.com','Paolillo99');

insert into Utente values ('LNDTMS99S13F138A','Thomas','Landi','13/11/1999','Via Nicola Farina','landithomas1999@gmail.com','tlandi3');

insert into Utente values ('SVRMRC79H19H703D','Marco','Savarese','19/06/1979','Via Salvador Allende','SavareseMarco1919@gmail.com','SavareseGranata19');

insert into Utente values ('CNCVLR01M44L483D','Valeria','Concilio','04/08/2001','Via Ferreria','ValeriaConcilio01@gmail.com','ValeriaC01');

insert into Utente values ('FRTFNC77L45L219Y','Francesca','Fortunato','05/07/1977','Via Machiavelli','FrancescaFortunato77@gmail.com','Fortunato77');

insert into Utente values ('MMLMTT85M14H199M','Matteo','Memoli','14/08/85','Via Giovanni Paolo II','MatteoMemoli85@gmail.com','Memoli85');

insert into Utente values ('CLPSRN94T65A944U','Serena','Colapesce','25/03/94','Via Don Mizzoni','SerenaColpasce94@gmail.com','Serena94');

insert into Utente values ('SNTGPP90T25A089N','Giuseppe','Santo','25/12/90','Via Trinità','SantoGiuseppe90@gmail.com','Santo90');

insert into Cellulare values ('3400427881','SPSGNR89T12G942K');

insert into Cellulare values ('3310378810','LNDRA00R41L781V');

insert into Cellulare values ('3331015300','PLLSN99C20H703T');

insert into Cellulare values ('3400721672','LNDTMS99S13F138A');

insert into Cellulare values ('3312564098','SVRMRC79H19H703D');

insert into Cellulare values ('3276509876','CNCVLR01M44L483D');

insert into Cellulare values ('3392555321','MMLMTT85M14H199M');

insert into Cellulare values ('3271467947','CLPSRN94T65A944U');

insert into Cellulare values ('3274422331','SNTGPP90T25A089N');

insert into Cellulare values ('089956743','SNTGPP90T25A089N');

insert into Cellulare values ('089955058','LNDTMS99S13F138A');

insert into Cellulare values ('3401234567','FRTFNC77L45L219Y');

insert into Pagamento values

('1234567891234567','Gennaro','Esposito','12/2025','123','SPSGNR89T12G942K');

insert into Pagamento values

('1223344556677889','Sara','Landi','10/2026','345','LNDSTRA00R41L781V');

insert into Pagamento values

('9887766554433221','Marco','Savarese','09/2023','567','SVRMRC79H19H703D');

insert into Pagamento values

('1029384756473829','Thomas','Landi','08/2025','789','LNDTMS99S13F138A');

insert into Pagamento values

('7968574635241324','Serena','Memoli','11/2024','910','PLLLSN99C20H703T');

insert into Pagamento values

('9164827391837464','Giuseppe','Santo','12/2024','101','SNTGPP90T25A089N');

insert into Pagamento values

('6826598196458178','Mattia','Pascal','08/2027','112','LNDTMS99S13F138A');

insert into Pagamento values

('9167591647657416','Francesca','Fortunato','03/2029','133','FRTFNC77L45L219Y');

insert into Pagamento values

('234567432rr34756','Matteo','Memoli','03/2022','919','MMLMTT85M14H199M');

insert into Pagamento values

('3242353453454543','Valeria','Concilio','01/2025','118','CNCVLR01M44L483D');

insert into Pagamento values

('9814545683746538','Serena','Colapesce','02/2023','113','CLPSRN94T65A944U');

insert into Pagamento values

('8956948763591365','Alessandro','Paolillo','07/2027','115','PLLLSN99C20H703T');

insert into Servizio

(Titolo,Descrizione,NumeroPosti,Costo,Data,Orainizio,Durata,Condiviso,Disciplina) values ('Corso di tennis', 'Corso di tennis per bambini con professionista certificato che si svolge in tre sedute', '10', '60', '15/05/2021', '16:00', '01:30', 'false', '2');

insert into Servizio

(Titolo,Descrizione,NumeroPosti,Costo,Data,Orainizio,Durata,Condiviso,Disciplina) values ('Prenotazione Struttura', 'Prenotazione campo per partita di calcio a 5', '1', '40', '17/05/2021', '19:00', '01:00', 'true', '9');

insert into Servizio

(Titolo,Descrizione,NumeroPosti,Costo,Data,Orainizio,Durata,Condiviso,Disciplina) values ('Lezione una-tantum golf', 'Lezione di golf erogato da professionista certificato', '15', '25', '20/06/2021', '14:00', '03:00', 'false', '4');

insert into Servizio

(Titolo,Descrizione,NumeroPosti,Costo,Data,Orainizio,Durata,Condiviso,Disciplina) values ('Consulenza', 'Migliorare la gestione del singolo atleta o tecnico', '1', '100', '26/05/2021', '17:00', '03:00', 'false', '5');

insert into Servizio

(Titolo,Descrizione,NumeroPosti,Costo,Data,Orainizio,Durata,Condiviso,Disciplina) values ('Lezione una-tantum Hip Hop', 'Corso di Hip Hop erogato da un professionista certificato', '15', '25', '05/08/2021', '10:00', '04:00', 'false', '7');

insert into Servizio

(Titolo,Descrizione,NumeroPosti,Costo,Data,Orainizio,Durata,Condiviso,Disciplina) values ('Prenotazione Struttura', 'Prenotazione campo per partita di calcio a 11', '1', '50', '14/05/2021', '20:00', '01:30', 'true', '8');

insert into Servizio

(Titolo,Descrizione,NumeroPosti,Costo,Data,Orainizio,Durata,Condiviso,Disciplina) values ('Corso di Pallavolo', 'Corso di Pallavolo erogato da un professionista certificato che si svolge in 5 sedute', '10', '100', '18/11/2021', '16:00', '02:00', 'false', '3');

insert into Servizio

(Titolo,Descrizione,NumeroPosti,Costo,Data,Orainizio,Durata,Condiviso,Disciplina) values ('Lezione una-tantum Arti Marziali', 'Lezione individuale di arti marziali con professionista certificato', '1', '35', '23/04/2021', '12:00', '01:30', 'false', '17');

insert into Servizio

(Titolo,Descrizione,NumeroPosti,Costo,Data,Orainizio,Durata,Condiviso,Disciplina) values ('Consulenza', 'Consulenza Sportiva per gli atleti di marcia, salto con l'asta, salto in alto', '1', '25', '01/07/2021', '16:00', '01:00', 'false', '28');

insert into Servizio

(Titolo,Descrizione,NumeroPosti,Costo,Data,Orainizio,Durata,Condiviso,Disciplina) values ('Prenotazione Struttura', 'Prenotazione campo al coperto per partita di pallavolo', '1', '40', '07/06/2021', '20:30', '01:30', 'true', '3');

insert into Servizio

(Titolo,Descrizione,NumeroPosti,Costo,Data,Orainizio,Durata,Condiviso,Disciplina) values ('Lezione una-tantum Scii', 'Lezione di Scii con professionista certificato per utenti alle prime armi', '1', '35', '22/12/2021', '15:00', '02:30', 'false', '24');

insert into Servizio

(Titolo,Descrizione,NumeroPosti,Costo,Data,Orainizio,Durata,Condiviso,Disciplina) values ('Consulenza', 'Consulenza sportiva per migliorare l'alimentazione durante le competizioni di canottaggio', '1', '55', '05/09/2021', '15:30', '01:30', 'false', '15');

insert into Servizio

(Titolo,Descrizione,NumeroPosti,Costo,Data,Orainizio,Durata,Condiviso,Disciplina) values ('Consulenza', 'Consulenza per migliorare l'elasticità dei muscoli e l'equilibrio interiore', '1', '25', '08/06/2021', '16:30', '01:00', 'false', '23');

insert into Servizio

(Titolo,Descrizione,NumeroPosti,Costo,Data,Orainizio,Durata,Condiviso,Disciplina) values ('Corso di nuoto', 'Corso di nuoto per principianti con professionista certificato che si svolge in 10 sedute', '5', '350', '06/01/2022', '17:00', '01:00', 'false', '14');

insert into Servizio

(Titolo,Descrizione,NumeroPosti,Costo,Data,Orainizio,Durata,Condiviso,Disciplina) values ('Prenotazione Struttura', 'Prenotazione campo di bocce al coperto', '1', '20', '13/10/2021', '10:00', '01:00', 'true', '16');

insert into Servizio

(Titolo,Descrizione,NumeroPosti,Costo,Data,Orainizio,Durata,Condiviso,Disciplina) values ('Corso di tennis', 'Corso di tennis per bambini con professionista certificato che si svolge in tre sedute', '10', '100', '14/05/2021', '16:00', '01:30', 'false', '2');

Insert into PropostaSocieta values ('1','12345678912');

Insert into PropostaSocieta values ('6','12341234912');

Insert into PropostaSocieta values ('10','12301010912');

Insert into PropostaSocieta values ('15','12345678912');
insert into PropostaSocieta values ('16','12345678912');

Insert into PropostaProfessionista values ('8','SBBMRC92M09H703Y');
Insert into PropostaProfessionista values ('9','BDAPPP86D23H703F');
Insert into PropostaProfessionista values ('2','SBBMRC92M09H703Y');
Insert into PropostaProfessionista values ('3','BDAPPP86D23H703F');

insert into Cooperazione values ('4','43215678912','DGRMRA99L01L219J');
insert into Cooperazione values ('5','12344321912','SNTMRA92T59B180C');
insert into Cooperazione values ('13','12301010912','MRRFNC82B45C351S');
insert into Cooperazione values ('7','43215678912','DGRMRA99L01L219J');
insert into Cooperazione values ('11','12341234912','SNTMRA92T59B180C');
insert into Cooperazione values ('12','12344321912','MRRFNC82B45C351S');
insert into Cooperazione values ('14','12301010912','MRRFNC82B45C351S');

insert into Struttura (Nome,Indirizzo,Descrizione) values ('Settembrino','via Settemebre','Via
settembre struttura che offre molti servizi');
insert into Struttura (Nome,Indirizzo,Descrizione) values ('De Gasperi','via Italia','Via Italia
Struttura per gli appassionati di sport');
insert into Struttura (Nome,Indirizzo,Descrizione) values ('Figliolia','via Famiglia','Via Famiglia
Struttura per chi ama stare insieme');
insert into Struttura (Nome,Indirizzo,Descrizione) values ('Olimpico','via Santi','Via Santi Struttura
per gli amanti dello sport');
insert into Struttura (Nome,Indirizzo,Descrizione) values ('Salvador','via Dei Mille','Via dei Mille
Struttura per gli amanti dle terzo tempo');

insert into Utilizzo values('2','1');
insert into Utilizzo values('6','2');
insert into Utilizzo values('10','3');
insert into Utilizzo values('15','4');
insert into Utilizzo values('1','5');
insert into Utilizzo values('3','4');
insert into Utilizzo values('7','2');
insert into Utilizzo values('11','1');
insert into Utilizzo values('14','5');
insert into Utilizzo values('5','3');

insert into Acquisto (Data,Servizio,Utente) values ('14/05/2021','1','SPSGNR89T12G942K');
insert into Acquisto (Data,Servizio,Utente) values ('14/05/2021','1','LNDRA00R41L781V');
insert into Acquisto (Data,Servizio,Utente) values ('14/05/2021','1','PLLSN99C20H703T');
insert into Acquisto (Data,Servizio,Utente) values ('14/05/2021','1','LNDTMS99S13F138A');
insert into Acquisto (Data,Servizio,Utente) values ('16/05/2021','2','SVRMRC79H19H703D');
insert into Acquisto (Data,Servizio,Utente) values ('19/06/2021','3','CNCVLR01M44L483D');
insert into Acquisto (Data,Servizio,Utente) values ('19/06/2021','3','FRTFNC77L45L219Y');
insert into Acquisto (Data,Servizio,Utente) values ('19/06/2021','3','MMLMTT85M14H199M');
insert into Acquisto (Data,Servizio,Utente) values ('19/06/2021','3','CLPSRN94T65A944U');

```

insert into Acquisto (Data,Servizio,Utente) values ('19/06/2021','3','SNTGPP90T25A089N');
insert into Acquisto (Data,Servizio,Utente) values ('19/06/2021','3','SPSGNR89T12G942K');
insert into Acquisto (Data,Servizio,Utente) values ('19/06/2021','3','LNDSRA00R41L781V');
insert into Acquisto (Data,Servizio,Utente) values ('19/06/2021','3','PLLSN99C20H703T');
insert into Acquisto (Data,Servizio,Utente) values ('19/06/2021','3','LNDTMS99S13F138A');
insert into Acquisto (Data,Servizio,Utente) values ('25/05/2021','4','LNDTMS99S13F138A');
insert into Acquisto (Data,Servizio,Utente) values ('04/08/2021','5','SVRMRC79H19H703D');
insert into Acquisto (Data,Servizio,Utente) values ('04/08/2021','5','CNCVLR01M44L483D');
insert into Acquisto (Data,Servizio,Utente) values ('04/08/2021','5','FRTFNC77L45L219Y');
insert into Acquisto (Data,Servizio,Utente) values ('11/05/2021','6','FRTFNC77L45L219Y');
insert into Acquisto (Data,Servizio,Utente) values ('17/11/2021','7','CLPSRN94T65A944U');
insert into Acquisto (Data,Servizio,Utente) values ('22/04/2021','8','SNTGPP90T25A089N');
insert into Acquisto (Data,Servizio,Utente) values ('30/06/2021','9','SPSGNR89T12G942K');
insert into Acquisto (Data,Servizio,Utente) values ('13/05/2021','1','SPSGNR89T12G942K');
insert into Acquisto (Data,Servizio,Utente) values ('18/06/2021','3','CLPSRN94T65A944U');
insert into Acquisto (Data,Servizio,Utente) values ('03/06/2021','10','LNDSRA00R41L781V');
insert into Acquisto (Data,Servizio,Utente) values ('21/12/2021','11','PLLSN99C20H703T');
insert into Acquisto (Data,Servizio,Utente) values ('04/09/2021','12','LNDTMS99S13F138A');
insert into Acquisto (Data,Servizio,Utente) values ('07/06/2021','13','SVRMRC79H19H703D');
insert into Acquisto (Data,Servizio,Utente) values ('05/01/2022','14','FRTFNC77L45L219Y');
insert into Acquisto (Data,Servizio,Utente) values ('09/10/2021','15','CLPSRN94T65A944U');

```

```

insert into Invito (Descrizione,DataEsito,Esito,Acquisto,UtenteInvitato) values ('Ciao Gennaro, ti
invito a partecipare alla partita di calcio a 5','16/05/2021','Accettato','5','SPSGNR89T12G942K');
insert into Invito (Descrizione,DataEsito,Esito,Acquisto,UtenteInvitato) values ('Ciao Sara, ti invito
a partecipare alla partita di calcio a 5','16/05/2021','Accettato','5','LNDSRA00R41L781V');
insert into Invito (Descrizione,DataEsito,Esito,Acquisto,UtenteInvitato) values ('Ciao Alessandro, ti
invito a partecipare alla partita di calcio a 5','16/05/2021','Rifiutato','5','PLLSN99C20H703T');
insert into Invito (Descrizione,DataEsito,Esito,Acquisto,UtenteInvitato) values ('Ciao Thomas, ti
invito a partecipare alla partita di calcio a 5','16/05/2021','Accettato','5','LNDTMS99S13F138A');
insert into Invito (Descrizione,DataEsito,Esito,Acquisto,UtenteInvitato) values ('Ciao Valeria, ti
invito a partecipare alla partita di calcio a 5','16/05/2021','Rifiutato','5','CNCVLR01M44L483D');
insert into Invito (Descrizione,DataEsito,Esito,Acquisto,UtenteInvitato) values ('Ciao Francesca, ti
invito a partecipare alla partita di calcio a 5','16/05/2021','Accettato','5','FRTFNC77L45L219Y');
insert into Invito (Descrizione,DataEsito,Esito,Acquisto,UtenteInvitato) values ('Ciao Matteo, ti
invito a partecipare alla partita di calcio a 5','16/05/2021','Rifiutato','5','MMLMTT85M14H199M');
insert into Invito (Descrizione,DataEsito,Esito,Acquisto,UtenteInvitato) values ('Ciao Gennaro, ti
invito a partecipare alla partita di calcio a 11','12/05/2021','Rifiutato','19','SPSGNR89T12G942K');
insert into Invito (Descrizione,DataEsito,Esito,Acquisto,UtenteInvitato) values ('Ciao Gennaro, ti
invito a partecipare alla partita di calcio a 11','12/05/2021','Accettato','19','SPSGNR89T12G942K');
insert into Invito (Descrizione,DataEsito,Esito,Acquisto,UtenteInvitato) values ('Ciao Sara, ti invito
a partecipare alla partita di calcio a 11','12/05/2021','Rifiutato','19','LNDSRA00R41L781V');
insert into Invito (Descrizione,Acquisto,UtenteInvitato) values ('Ciao Alessandro, ti invito a
partecipare alla partita di calcio a 11','19','PLLSN99C20H703T');
insert into Invito (Descrizione,DataEsito,Esito,Acquisto,UtenteInvitato) values ('Ciao Thomas, ti
invito a partecipare alla partita di calcio a 11','13/05/2021','Accettato','19','LNDTMS99S13F138A');

```

```

insert into Invito (Descrizione,DataEsito,Esito,Acquisto,UtenteInvitato) values ('Ciao Marco, ti
invito a partecipare alla partita di calcio a
11','13/05/2021','Accettato','19','SVRMRC79H19H703D');
insert into Invito (Descrizione, Acquisto, UtenteInvitato) values ('Ciao Francesca, ti invito a
partecipare alla partita di calcio a 11','19','FRTFNC77L45L219Y');
insert into Invito (Descrizione,DataEsito,Esito,Acquisto,UtenteInvitato) values ('Ciao Serena, ti
invito a partecipare alla partita di palavolo','3/06/2021','Rifiutato','25','CLPSRN94T65A944U');
insert into Invito (Descrizione,DataEsito,Esito,Acquisto,UtenteInvitato) values ('Ciao Valeria, ti
invito a partecipare alla partita di palavolo','4/06/2021','Accettato','25','CNCVLR01M44L483D');
insert into Invito (Descrizione,DataEsito,Esito,Acquisto,UtenteInvitato) values ('Ciao Gennaro, ti
invito a partecipare alla partita di bocce','09/10/2021','Rifiutato','30','SPSGNR89T12G942K');
insert into Invito (Descrizione,DataEsito,Esito,Acquisto,UtenteInvitato) values ('Ciao Sara, ti invito
a partecipare alla partita di bocce','10/10/2021','Accettato','30','LNDSTRA00R41L781V');
insert into Invito (Descrizione, Acquisto, UtenteInvitato) values ('Ciao Alessandro, ti invito a
partecipare alla partita di bocce','30','PLLSN99C20H703T');

```

```

Insert into Preferenza (CF,Livello) values ('PLLSN99C20H703T','Amatoriale');
Insert into Preferenza (CF,Livello) values ('LNDTMS99S13F138A','Agonistico');
Insert into Preferenza (CF,Livello) values ('PLLSN99C20H703T','Novizio');
Insert into Preferenza (CF,Livello) values ('LNDTMS99S13F138A','Amatoriale');
Insert into Preferenza (CF,Livello) values ('CNCVLR01M44L483D','Amatoriale');
Insert into Preferenza (CF,Livello) values ('SVRMRC79H19H703D','Novizio');
Insert into Preferenza (CF,Livello) values ('FRTFNC77L45L219Y','Amatoriale');
Insert into Preferenza (CF,Livello) values ('CNCVLR01M44L483D','Agonistico');
Insert into Preferenza (CF,Livello) values ('SVRMRC79H19H703D','Agonistico');
Insert into Preferenza (CF,Livello) values ('MMLMTT85M14H199M','Novizio');
Insert into Preferenza (CF,Livello) values ('CNCVLR01M44L483D','Amatoriale');
Insert into Preferenza (CF,Livello) values ('CLPSRN94T65A944U','Agonistico');
Insert into Preferenza (CF,Livello) values ('SVRMRC79H19H703D','Agonistico');
Insert into Preferenza (CF,Livello) values ('SNTGPP90T25A089N','Amatoriale');
Insert into Preferenza (CF,Livello) values ('MMLMTT85M14H199M','Novizio');
Insert into Preferenza (CF,Livello) values ('FRTFNC77L45L219Y','Amatoriale');
Insert into Preferenza (CF,Livello) values ('SNTGPP90T25A089N','Amatoriale');
Insert into Preferenza (CF,Livello) values ('CLPSRN94T65A944U','Novizio');
Insert into Preferenza (CF,Livello) values ('MMLMTT85M14H199M','Agonistico');
Insert into Preferenza (CF,Livello) values ('FRTFNC77L45L219Y','Agonistico');
Insert into Preferenza (CF,Livello) values ('SNTGPP90T25A089N','Amatoriale');
Insert into Preferenza (CF,Livello) values ('CLPSRN94T65A944U','Novizio');

```

7. Query SQL

Workpackage	Task	Responsabile
WP3	SQL: Query	Memoli Matteo

7.1. Query con operatore di aggregazione e join: Elenco riassuntivo dei servizi offerti

Visualizza l'elenco delle discipline offerte dalle società e dai professionisti mostrando le loro informazioni, il prezzo minimo, medio e massimo relativo a quella disciplina

```

Select Societa.Denominazione as NomeFornitore, Societa.telefono as TelefonoFornitore, Societa.Indirizzo as IndirizzoS, Societa.Cap as Cap,
       Disciplina.nome as Disciplina, count(*) as Quantita, min(Servizio.costo) as Minimo,
       max(servizio.costo) as Massimo, round((avg(Servizio.Costo)),2) as Media
From Societa join PropostaSocieta on (Societa.CF= PropostaSocieta.Societa)
       join Servizio on(PropostaSocieta.Servizio = Servizio.Codice)
       join Disciplina on (Servizio.Disciplina = disciplina.id)
Group by NomeFornitore, TelefonoFornitore,IndirizzoS, Cap, Disciplina,nome
Union
Select Professionista.Cognome as NomeFornitore, Professionista.Telefono as TelefonoFornitore, Professionista.Indirizzo as IndirizzoS, Professionista.Cap as Cap,
       Disciplina.nome as Disciplina, count(*) as Quantita, min(Servizio.costo) as Minimo,
       max(servizio.costo) as Massimo, round((avg(Servizio.Costo)),2) as Media
From Professionista join PropostaProfessionista on (Professionista.CF= PropostaProfessionista.Professionista)
       join Servizio on(PropostaProfessionista.Servizio = Servizio.Codice)
       join Disciplina on (Servizio.Disciplina = disciplina.id)
Group by NomeFornitore, TelefonoFornitore,IndirizzoS, Cap, Disciplina,nome

```

7.2. Query nidificata complessa: Utenti Invitati

Visualizza l'elenco degli utenti invitati che hanno accettato l'invito di 'Francesca Fortunato' per il servizio di 'Calcio a 11' nella data sopra indicata

```

SELECT Username as UtenteInvitato
FROM Utente
where exists (select * from Invito, Acquisto, Servizio, Disciplina, Utente U
              where Invito.UtenteInvitato = Utente.CF AND Invito.Acquisto = Acquisto.ID
              AND Acquisto.Servizio = Servizio.Codice AND Servizio.disciplina = disciplina.ID
              AND U.CF = Acquisto.Utente AND disciplina.nome = 'Calcio a 11' AND Servizio.data = '2021-05-14'
              AND U.Nome ='Francesca' AND U.Cognome = 'Fortunato' AND Invito.Esito = 'Accettato');

```

7.3. Query insiemistica: Tutte le società che non offrono il calcio a 11

La query visualizza tutte le Società e professionisti che non offrono il Servizio di Calcio.

```

Select distinct Denominazione
From Societa
where Denominazione not in --Poi trovo tutte quelle che offrono il calcio
    (Select s1.denominazione
     From PropostaSocieta ps join Servizio s on (s.codice = ps.servizio)
                          join Societa s1 on (s1.cf=ps.societa)
                          join Disciplina D on (D.ID = S.Disciplina)
                          where D.nome like 'Calcio a %')

UNION
Select Cognome
from Professionista
where Cognome not in --Poi trovo tutte quelle che offrono il calcio
    (Select pl.cognome
     From PropostaProfessionista ps join Servizio s on (s.codice = ps.servizio)
                                   join Professionista pl on (pl.cf=ps.professionista)
                                   join Disciplina D on (D.ID = S.Disciplina)
                                   where D.nome like 'Calcio a %')

```

7.4. Altre Query

7.4.1. Società che offrono almeno 3 servizi ma nessuno di calcio

La query permette di visualizzare le Società che offrono almeno 3 servizi e, allo stesso tempo, la disciplina coinvolta non riguarda il "Calcio".

```

Select SN.Denominazione from(
    Select S.denominazione, count(*) as NumServizi
    FROM Societa S, PropostaSocieta PS, Servizio Se
    Where S.CF=PS.Societa AND PS.Servizio = Se.codice
    Group By S.denominazione) as SN
WHERE SN.NumServizi >= 3

INTERSECT
Select distinct Denominazione
From Societa
where Denominazione not in --Poi trovo tutte quelle che offrono il calcio
    (Select s1.denominazione
     From PropostaSocieta ps join Servizio s on (s.codice = ps.servizio)
                          join Societa s1 on (s1.cf=ps.societa)
                          join Disciplina D on (D.ID = S.Disciplina)
                          where D.nome like 'Calcio a %')

```


8. Viste

<i>Workpackage</i>	<i>Task</i>	<i>Responsabile</i>
WP4	Viste	Paolillo Alessandro

8.1. Vista Voga

Creiamo la vista che permette di visualizzare le discipline e il numero di vendite delle discipline coinvolte nell'acquisizione

```
Create view Voga as
SELECT disciplina.nome, count(disciplina.nome) as numVendite
FROM Acquisto, Servizio, Disciplina
WHERE Acquisto.servizio = servizio.codice AND servizio.disciplina=disciplina.id
Group by disciplina.nome
```

8.1.1. Query con Vista: Voga

Con l'aiuto della vista 'Voga' creata possiamo selezionare la disciplina più in voga del momento.

```
Select nome, numVendite
From Voga
Where numVendite = (Select max(numVendite)
                    From Voga)
```

8.2. Vista NumeroAcquisti

Creiamo la vista che permette di visualizzare il numero degli acquisti di ogni utente.

```
create view NumeroAcquisti as
select A.Utente,U.Nome,U.Cognome,count(*) as QuantitaAcquistate
from Acquisto A join Utente U on (U.CF = A.Utente)
group by A.Utente,U.Nome, U.Cognome
```

8.2.1. Query con Vista: NumeroAcquisti

Con l'aiuto della vista 'NumeroAcquisti' possiamo selezionare chi ha acquistato più servizi in questo momento

```
select * from NumeroAcquisti
where QuantitaAcquistate = (select max(QuantitaAcquistate)
                           from NumeroAcquisti)
```

8.3. Vista: Professionisti Cooperanti

Definisco la vista per visualizzare le informazioni dei professionisti e le società con cui cooperano.

```
Create View ProfCooperanti (CognomeProf, NomeProf, CellProf, NomeSocieta) as
Select distinct p.Cognome, p.Nome, p.Telefono, s.denominazione
From professionista p join cooperazione c on p.CF=c.professionista
    join societa s on c.societa=s.CF;
```

8.3.1. Query con vista: Cooperazioni professionista Morriello

Visualizzo, utilizzando la vista appena creata, le società con cui coopera il professionista Morriello

```
Select NomeSocieta
from ProfCooperanti
where CognomeProf='Morriello';
```

8.3.1.1. Sottovista: NumeroProf

Definisco la sottovista "NumeroProf" per mostrare il numero di professionisti che cooperano con una società.

```
Create View NumeroProf as
Select NomeSocieta, count(CognomeProf) as NumeroProfessionisti
From ProfCooperanti
group by NomeSocieta;
```

8.3.1.1.1. Query con Sottovista: NumeroProf

Visualizzo le società che collaborano con un solo professionista.

```
select NomeSocieta
from NumeroProf
where NumeroProfessionisti=1;
```

9. Trigger

9.1. Trigger inizializzazione: *Controllo Username*

Workpackage	Task	Responsabile
WP1	Trigger inizializzazione/popoloamento database	Savarese Marco

9.1.1. Creazione trigger controllo username_utente

Il seguente trigger consente di verificare se l'username del nuovo utente che si sta inserendo è già utilizzato da un professionista o una società.

```
DROP FUNCTION Check_username_utente() cascade;
CREATE FUNCTION Check_username_utente() returns trigger as $$
BEGIN
if (exists(select*from Societa where username =new.username) or exists(select*from Professionista
where username =new.username)) then
raise exception 'Username esistente';
end if;
Return new;
END
$$ Language plpgsql;
```

```
CREATE TRIGGER check_insert_usernameUtente
BEFORE INSERT ON UTENTE
FOR EACH ROW EXECUTE PROCEDURE Check_username_utente();
```

9.1.2. Creazione trigger controllo username_professionista

Il seguente trigger consente di verificare se l'username del nuovo Professionista che si sta inserendo è già utilizzato da un utente o una società.

```
DROP FUNCTION Check_username_Professionista() cascade;
CREATE FUNCTION Check_username_Professionista() returns trigger as $$
BEGIN
if (exists(select*from Societa where username =new.username) or exists(select*from Utente where
username =new.username)) then
raise exception 'Username esistente';
end if;
Return new;
END
$$ Language plpgsql;
```

```
CREATE TRIGGER check_insert_usernameProfessionista
BEFORE INSERT ON Professionista
FOR EACH ROW EXECUTE PROCEDURE Check_username_Professionista();
```

9.1.3. Creazione trigger username_societa

Il seguente trigger consente di verificare se l'username della nuova societa che si sta inserendo è già utilizzato da un professionista o da un utente.

```
DROP FUNCTION Check_username_Societa() cascade;
CREATE FUNCTION Check_username_Societa() returns trigger as $$
BEGIN
if (exists(select * from Utente where username = new.username) or exists(select * from Professionista
where username = new.username)) then
raise exception 'Username esistente';
end if;
Return new;
END
$$ Language plpgsql;

CREATE TRIGGER check_insert_usernameSocieta
BEFORE INSERT ON Societa
FOR EACH ROW EXECUTE PROCEDURE Check_username_Societa();
```

9.1.4. Creazione trigger Inserimenti in PropostaSocieta dato inserimenti in Cooperazione

Visto e considerato che un servizio svolto in cooperazione tra una Societa e un Professionista deve essere fornito da una Societa, realizziamo il seguente trigger affinché un inserimento di un servizio in Cooperazione venga messo automaticamente anche in PropostaSocieta.

```
DROP FUNCTION insert_PropostaSocieta() cascade;
CREATE FUNCTION insert_PropostaSocieta() returns trigger as $$
BEGIN
select Servizio, Societa into new.Servizio, new.Societa from newTab;
INSERT into PropostaSocieta values (new.Servizio, new.Societa);
RETURN NULL;
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER Popolamento_PropostaSocieta
AFTER INSERT ON Cooperazione
REFERENCING NEW TABLE as newTab FOR EACH STATEMENT
EXECUTE PROCEDURE insert_PropostaSocieta();
```

9.2. Trigger per vincoli aziendali

Workpackage	Task	Responsabile
WP4	Trigger per vincoli aziendali	Paolillo Alessandro

9.2.1. Trigger1: Creazione trigger controllo acquisto

Il trigger monitora l'inserimento di un nuovo acquisto. Quest'ultimo, infatti, deve essere possibile solo se la data di inizio del servizio che si desidera acquistare è successiva alla data di esecuzione dell'acquisto. Il trigger, inoltre, deve necessariamente verificare che il numero di acquisti inseriti per quel determinato servizio sia minore o uguale al numero di posti disponibili per quel servizio.

```

DROP FUNCTION check_Acquisti() cascade;
Create function check_Acquisti() RETURNS trigger as $$
Begin
  if(((Select Servizio.numeroposti
        from Servizio
        where new.servizio=Servizio.codice)=(Select count(*) from Acquisto where Acquisto.servizio
= new.servizio)) and ((Select Servizio.data  from Servizio where new.servizio=Servizio.codice) <=
new.Data)) then raise Exception 'Numero posti terminati e servizio scaduto';
    end if;
  if((Select Servizio.numeroposti
        from Servizio
        where new.servizio=Servizio.codice)=(Select count(*) from Acquisto where Acquisto.servizio
= new.servizio)) then raise Exception 'Numero posti Terminati';
    end if;
  if((Select Servizio.data
        from Servizio
        where new.servizio=Servizio.codice) <= new.Data) then raise Exception 'Servizio Scaduto';
    end if;
RETURN new;
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER check_inserimentoAcquisto
BEFORE INSERT ON ACQUISTO
FOR EACH ROW EXECUTE procedure check_Acquisti();

```

9.2.2. Trigger2: Creazione trigger controllo invito condivisibile

Il trigger è usato per controllare che il servizio coinvolto in un invito sia condivisibile.

```
DROP FUNCTION check_Invito() cascade;
Create function check_Invito() RETURNS trigger as $$
Begin
if((SELECT Servizio.Condiviso From Acquisto, Servizio Where new.acquisto=acquisto.id AND
acquisto.servizio=Servizio.codice)='false')
then raise Exception 'Servizio non codivisibile';
end if;
RETURN new;
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER check_inserimentoInvito
BEFORE INSERT ON Invito
FOR EACH ROW EXECUTE procedure check_Invito();
```