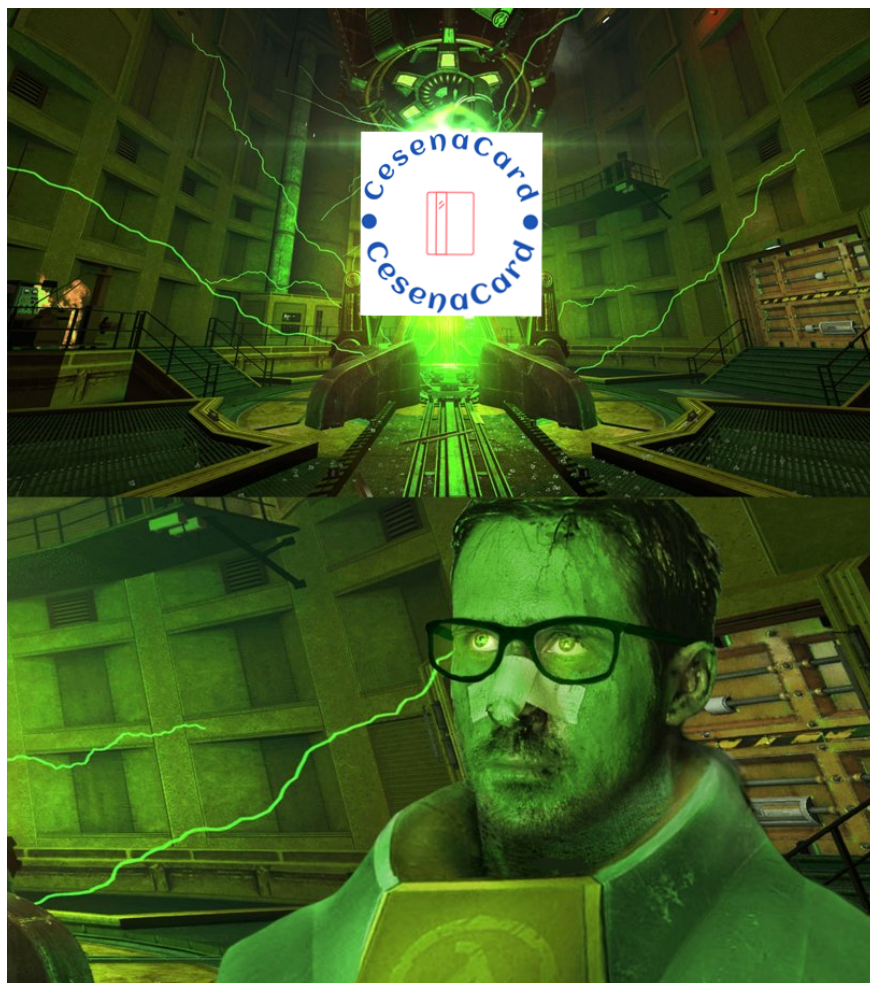


Relazione Basi di Dati

Dobrianskiy, Sergio
`sergio.dobrianskiy@studio.unibo.it`
0001019553

Rigato, Valentina
`valentina.rigato@studio.unibo.it`
0001027266

01 Marzo 2024



Esempio

Contents

1	Introduzione	6
2	Analisi dei requisiti	6
2.1	Requisiti in linguaggio naturale	6
2.2	Estrazione dei concetti fondamentali	7
2.2.1	Entità e Attributi	7
2.2.2	Analisi dettagliata di entità e attributi	10
2.2.3	Funzionalità richieste dall'applicativo	12
2.3	Schema Iniziale	13
2.4	Schema scheletro prima delle correzioni	14
2.4.1	User	14
2.4.2	Ente	15
2.4.3	Eventi	16
2.4.4	CityCard	17
2.4.5	Carta di credito	17
2.4.6	Sezione delle news	17
2.4.7	Abbonamento	18
2.4.8	Sconti	18
3	Progettazione Logica	18
3.1	Stima del volume dei dati	18
3.2	Operazioni richieste e la loro frequenza	19
3.3	Frequenza e costo degli accessi	21
3.4	Analisi delle ridondanze	21
3.5	Traduzioni delle operazioni in Query SQL	21
3.5.1	Recuperare tutti gli utenti	21
3.5.2	Recuperare tutti gli enti con e senza recensioni	22
3.5.3	Recupero lista eventi	22
3.5.4	Recupero lista servizi	23
3.5.5	Recupero acquisti di un utente	23
3.5.6	Reitorna statistiche amministratore	24
3.5.7	Recupero statistiche fornitore	24
3.5.8	Recupero listino abbonamenti	25
3.5.9	Creazione nuovo ente	25
3.5.10	Inserimento voto utente	25
3.5.11	Reset recensioni di un ente	26
3.5.12	Creazione servizio	26
3.5.13	Registrazione nuovo utente	26
3.5.14	Registrazione nuova carta di credito	26
3.5.15	Creazione nuova CityCard	27
3.5.16	Disattivazione CityCard	27
3.5.17	Rendere predefinita una carta di credito	27
3.5.18	Sottoscrizione abbonamento	27
3.5.19	Associa utente a ente	28

3.5.20	Effettua check-in	28
3.5.21	Ottieni CityCard di un utente	28
3.5.22	Partecipa a un evento	28
3.5.23	Edita Utente	28
3.5.24	Banna/sbanna utente	28
3.5.25	Cancella carta di credito	29
3.5.26	Effettua login	29
3.5.27	Ritorna, se presente, la CityCard attiva dell'utente	29
3.5.28	Ritorna ente associato all'utente	29
3.5.29	ritorna, se presente, la sottoscrizione attiva attiva dell'utente	30
3.5.30	Ritorna lista check-in fatti da un utente	30
3.5.31	Recupera enti	30
3.5.32	Recupera carte di credito di un utente	30
3.5.33	Compra servizio	30
3.5.34	Creazione evento periodico	31
3.5.35	Creazione evento occasionale	32

4 Progetto dello schema concettuale 32

1 Introduzione

L'obiettivo del progetto è di realizzare una piattaforma di registrazione ad un servizio di acquisto di una carta elettronica, la CityCard. CityCard è un sistema che favorisce l'interazione tra turisti e fornitori di servizi. I tipi di servizi possono essere diversi.

Per accedere al sito web i clienti dovranno creare un account fornendo le loro generalità. Una volta registrato l'utente base potrà comprare una CityCard che avrà una validità limitata. Ogni CityCard avrà associato un cliente, una carta di credito e un saldo che potrà essere usato per comprare e attivare servizi.

Di default la CityCard permette l'utilizzo del servizio di trasporto pubblico urbano. Inoltre potrà venire utilizzata per comprare servizi forniti dai fornitori di vario genere come visite guidate ai musei, biglietti utilizzabili sui trasporti extra urbani.

All'interno del portale web il cliente potrà vedere in homepage una lista di eventi organizzati dai fornitori. Gli eventi potranno essere dei semplici annunci di servizio oppure dei periodi di tempo nei quali potranno essere acquistabili servizi di una certa categoria con uno sconto. Potrà anche visualizzare la lista di fornitori con i relativi servizi.

Si prevede l'esistenza di tre tipi di account: cliente, fornitore, admin.

L'utente potrà vedere la parte pubblica del servizio, consultare i servizi disponibili, avrà la possibilità di lasciare una recensione per il servizio usufruito.

Gli account dei fornitori potranno invece rendere disponibili servizi, visualizzare statistiche relative ai propri servizi tra cui gli acquisti.

Gli account degli amministratori avranno accesso completo al database e potranno gestire gli utenti e accedere a statistiche.

2 Analisi dei requisiti

2.1 Requisiti in linguaggio naturale

La seguente descrizione riporta in linguaggio naturale i requisiti per il nostro sistema informativo:

"La software house CityNet© richiede un sistema informativo per la gestione del servizio CityCard che consenta agli utenti di acquistare abbonamenti e accedere a servizi ed eventi forniti da enti affiliati. CityCard può essere emessa sia in formato fisico che virtuale (online) e ha una durata di 5 anni, l'emissione della carta è gratuita e ogni utente ne può avere attiva al massimo una.

Gli abbonamenti disponibili con CityCard si suddividono in tre tipologie, ciascuna con una durata e un prezzo specifici. L'abbonamento permette agli utenti di acquistare servizi o prenotare eventi forniti dagli enti partner con uno sconto che varia dalla durata dell'abbonamento.

Il servizio deve essere fornito su un sito al quale l'utente si deve registrare.

Gli utenti si suddividono in tre categorie: cliente, fornitore e admin o amministratore. Una persona può creare più di un account.

Gli enti sono gestiti da utenti di tipo "fornitore", i quali, una volta associati a un ente, hanno la possibilità di creare nuovi servizi o eventi.

Al primo login gli utenti di tipo "cliente" possono solamente ottenere una CityCard e gestire le carte di credito. Dopo l'attivazione potranno accedere anche alla sezione abbonamenti. Una volta sottoscritto all'abbonamento potranno accedere a tutti i servizi.

I clienti potranno aggiungere più di una carta di credito e renderne una predefinita per i pagamenti.

I clienti possono acquistare i servizi forniti utilizzando la carta di credito predefinita e godendo dello sconto già nominato in precedenza. Inoltre la CityCard permette di partecipare agli eventi gratuiti e di usufruire gratuitamente del trasporto pubblico. I clienti, inoltre, avranno la possibilità di lasciare una recensione sul servizio acquistato o sull'evento a cui ha partecipato.

L'utente di tipo "admin" ha la facoltà di bannare gli altri utenti. I clienti, invece, possono acquistare i servizi offerti e prenotare eventi attraverso il sistema.

Il sistema dovrà essere progettato per garantire una gestione efficace delle diverse funzionalità descritte, assicurando un'esperienza utente intuitiva e fluida per tutte le tipologie di utenti."

2.2 Estrazione dei concetti fondamentali

2.2.1 Entità e Attributi

Lo schema va analizzato per individuarne le parole e le espressioni chiave, con queste verrà realizzato un primo schema riassuntivo che verrà raffinato in seguito. I termini di rilievo appaiono nel testo con un colore diverso:

"La software house CityNet© richiede un sistema informativo per la gestione del servizio **CityCard** che consenta agli utenti di acquistare **abbonamenti** e accedere a **servizi** ed **eventi** forniti da enti affiliati. CityCard può essere emessa sia in formato fisico che virtuale (online) e ha una durata di 5 anni, l'emissione della carta è gratuita e ogni utente ne può avere attiva al massimo una.

Gli abbonamenti disponibili con CityCard si suddividono in tre **tipologie**, ciascuna con una durata e un prezzo specifici. L'abbonamento permette agli utenti di acquistare servizi o prenotare eventi forniti dagli **enti** partner con uno sconto che varia dalla durata dell'abbonamento.

Il servizio deve essere fornito su un **sito** al quale l'**utente** si deve registrare.

Gli utenti si suddividono in tre categorie: **cliente**, **fornitore** e **admin** o amministratore. Una persona può creare più di un account.

Gli enti sono gestiti da utenti di tipo "fornitore", i quali, una volta associati a un ente, hanno la possibilità di creare nuovi servizi o eventi.

Al primo login gli utenti di tipo "cliente" possono solamente ottenere una CityCard e gestire le carte di credito. Dopo l'attivazione potranno accedere anche alla sezione abbonamenti. Una volta sottoscritto all'abbonamento potranno accedere a tutti i servizi.

I clienti potranno aggiungere più di una [carta di credito](#) e renderne una predefinita per i pagamenti.

I clienti possono acquistare i servizi forniti utilizzando la carta di credito predefinita e godendo dello [sconto](#) già nominato in precedenza. Inoltre la CityCard permette di partecipare agli eventi gratuiti e di usufruire gratuitamente del trasporto pubblico. I clienti, inoltre, avranno la possibilità di lasciare una [recensione](#) sul servizio acquistato o sull'evento a cui ha partecipato.

L'utente di tipo "admin" ha la facoltà di [bannare](#) gli altri utenti. I clienti, invece, possono acquistare i servizi offerti e prenotare eventi attraverso il sistema.

Il sistema dovrà essere progettato per garantire una gestione efficace delle diverse funzionalità descritte, assicurando un'esperienza utente intuitiva e fluida per tutte le tipologie di utenti."

La descrizione del prodotto è già abbastanza chiara, ma presenta alcune entità che vengono nominate più di una volta usando dei sinonimi, sarà quindi importante chiarire che si sta parlando sempre della stessa entità e non confondersi.

Table 1: Estrazione delle entità principali

Termine	Sinonimi usati	Descrizione
CityCard	Carta servizi	x Tessera che permette l'acquisto di un abbonamento
Clienti	Visitatori	Persone che usufruiscono di servizi offerti da un ente
Fornitori	Venditore, rifornitore	Persone che, per conto di terzi, fornisce servizi per la clientela
Admin	Amministratore	Persona responsabile della gestione del sito
Servizi	Attività, prestazioni	Operazioni svolte per soddisfare le esigenze dei clienti
Abbonamento	Sottoscrizione	Contratto che prevede un pagamento una tantum per poter accedere ai servizi
Ente	Associato	Un'azienda o un'entità con accordi per offrire servizi

Continued on next page

Table 1: Estrazione delle entità principali (Continued)

Termine	Sinonimi usati	Descrizione
CityCard	Carta servizi	x Tessera che permette l'acquisto di un abbonamento
Eventi	Incontri	Attività organizzate per riunire persone in occasione specifiche
Recensioni	Valutazioni	Commenti o giudizi espressi dai clienti su servizi
Sconto	Riduzione	Ribasso del prezzo originale di un prodotto o servizio.

Lo schema concettuale nella sua versione finale si avvarrà delle seguenti entità e associazioni:

Table 2: Entità e associazioni

Nome	Tipo		Descrizione
Utente	E		Possessore di un account
Servizio	E		Attività messa a disposizione da un fornitore
Ente	E		Associazioni che forniscono servizi (es. Musei)
Evento	E		Avvenimento gratuito con o senza prenotazione
Cliente	E		Colui che potrà usufruire di servizi
Fornitore	E		Erogatore servizi o eventi
Transazione	A		Acquisto di un servizio
Recensione	E		Valutazione di un servizio, evento o fornitore
Carta di credito	E		Mezzo con il quale si effettua la transazione
Possiede Card	City	A	Lega l'utente alla city card a lui assegnata
Possiede Credito	Carta	A	Lega l'utente alla sua carta di credito
Abbonamento	E		Permette l'utilizzo del trasporto pubblico

Continued on next page

Table 2: Entità e associazioni (Continued)

Nome	Tipo	Descrizione
Utente	E	Possessore di un account
Organizza	A	Attività che lega il fornitore all'utente garantendo servizi o eventi
Lavora	A	Indica per quale/i ente/i collabora il fornitore
Admin	E	Utente che ha il controllo della piattaforma
Partecipa	A	I clienti partecipano agli eventi
Sottoscrive	A	Attività di un utente che si "abbona" ad uno o più fornitori per seguire le novità
Abbona	A	Un cliente si abbona ad un abbonamento
CityCard	E	Carta che permette di usufruire dei servizi
Percentuale sconto	A	L'esatto importo di riduzione di prezzo
Sconti	E	Riduzione di prezzo di un servizio
Prenotazione	A	Attività che lega un utente e uno o più eventi laddove è necessaria confermare la propria presenza
Trasporti pubblici*	E	Lista dei trasporti da interrogare
Viaggia	A	Lista utilizzi CityCard

2.2.2 Analisi dettagliata di entità e attributi

Utente: L'utente sarà l'utilizzatore generico del servizio e al database al quale sarà appoggiato. Ogni utente dovrà appartenere ad una delle seguenti categorie.

- Cliente: può visionare e comprare i servizi offerti
- Fornitore: può creare e offrire servizi
- Admin: ha accesso totale al database e può compiere azioni necessarie per la moderazione e la manutenzione del sistema.

Le informazioni minime associate a ciascun utente saranno:

- Nome

- Cognome
- Codice fiscale
- email

Ente: L'ente sarà l'azienda partner che metterà a disposizione i suoi servizi e gli eventi organizzati in favore dei clienti. Le informazioni minime associate a ciascun ente saranno:

- Nome
- Descrizione dell'ente
- La data di creazione

Servizio: Il servizio è la macro area che il fornitore mette a disposizione nella piattaforma. Un servizio avrà queste caratteristiche.

- Fornitore: sarà l'ente che creerà il servizio

Evento: Con evento si intende l'attività specifica all'interno del servizio. Ad esempio, si può acquistare

- Prezzo: il costo per poterne usufruire
- Data di validità: data in cui potrà usufruire dell'evento o il range di date disponibili

Abbonamento: L'abbonamento è una sottoscrizione tramite la quale il cliente potrà accedere alla lista eventi e a quella dei servizi per poterli prenotare o acquistare. Ci sono tre tipi di abbonamento tra cui scegliere, tutte con una durata e un prezzo specifico. Un abbonamento avrà le seguenti caratteristiche:

- Descrizione del tipo di abbonamento
- Prezzo
- Periodo di validità

CityCard: La CityCard è una tessera fisica o virtuale che permette la sottoscrizione di un abbonamento. Un utente può avere solo una CityCard alla volta ed avrà la durata massima di 5 anni. Una CityCard avrà le seguenti caratteristiche:

- Data di emissione
- Data di scadenza
- Numero identificativo della tessera

Ban di un utente: La possibilità di bannare un utente è prerogativa dell'admin che, a sua discrezione, può intercedere un utente dall'utilizzo della piattaforma. Se l'utente viene bannato non gli è più possibile fare il login. Caratteristiche di questa associazione:

- Essere un admin
- Il cliente e il fornitore hanno un attributo booleano "bannato" il cui valore cambia in base allo stato in cui si trova (attivo o disattivato).

Carta di credito: La carta di credito è lo strumento con la quale si può acquistare un abbonamento, ogni cliente può avere più carte di credito associate al suo account ma solo una può essere predefinita per gli acquisti. Le caratteristiche della carta di credito sono le seguenti:

- Numero della carta di credito
- Mese e anno di scadenza
- La data di scadenza
- Un attributo che chiarisce quale carta di credito è quella di default

2.2.3 Funzionalità richieste dall'applicativo

Di seguito un elenco delle funzionalità richieste per l'cliente:

- c1) Richiedere una CityCard
- c2) Sottoscrivere un abbonamento per una CityCard
- c3) Aggiungere carte di credito al proprio account
- c4) Rendere predefinita una carta di credito
- c5) Acquistare un servizio
- c6) Prenotare un evento
- c7) Lasciare rating ai servizi acquistati (1-5 stelle)
- c8) Effettuare checkin per il mezzo di trasporto scelto

Di seguito un elenco delle funzionalità richieste per un amministratore:

- a1) Visualizzare lista utenti
- a2) Bannare tutti i tipi di account
- a3) Cancellare enti

a4) Resetare recensioni di un ente

a5) Consultare le statistiche

Di seguito un elenco delle funzionalità richieste per un fornitore:

b1) Creare enti

b2) Associarsi a un ente

b3) Creare servizi

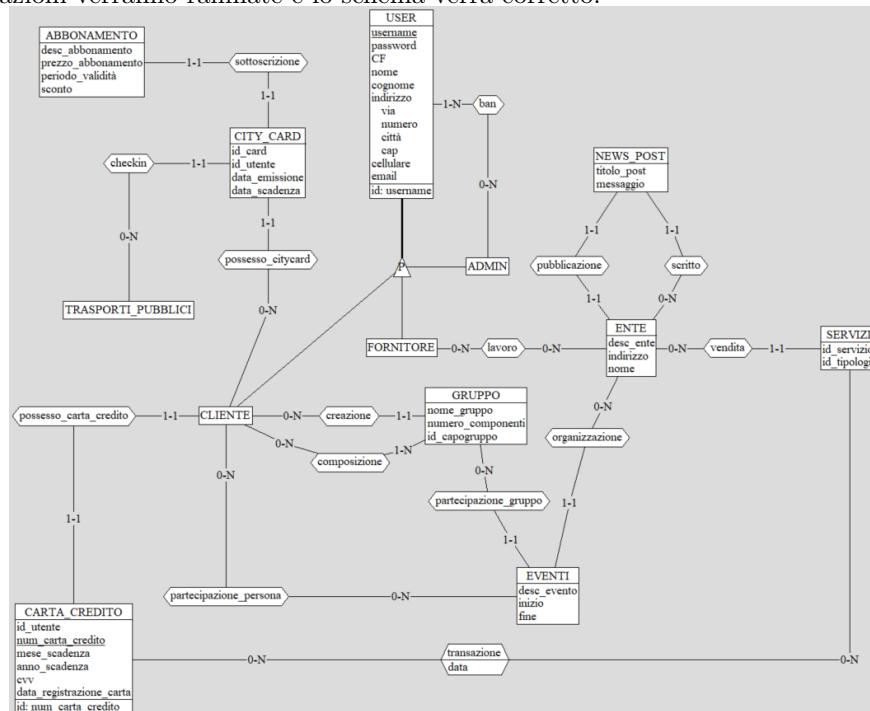
b4) Creare eventi

b5) Consultare statistiche riguardo l'ente al quale è associato

b6) Consultare il saldo

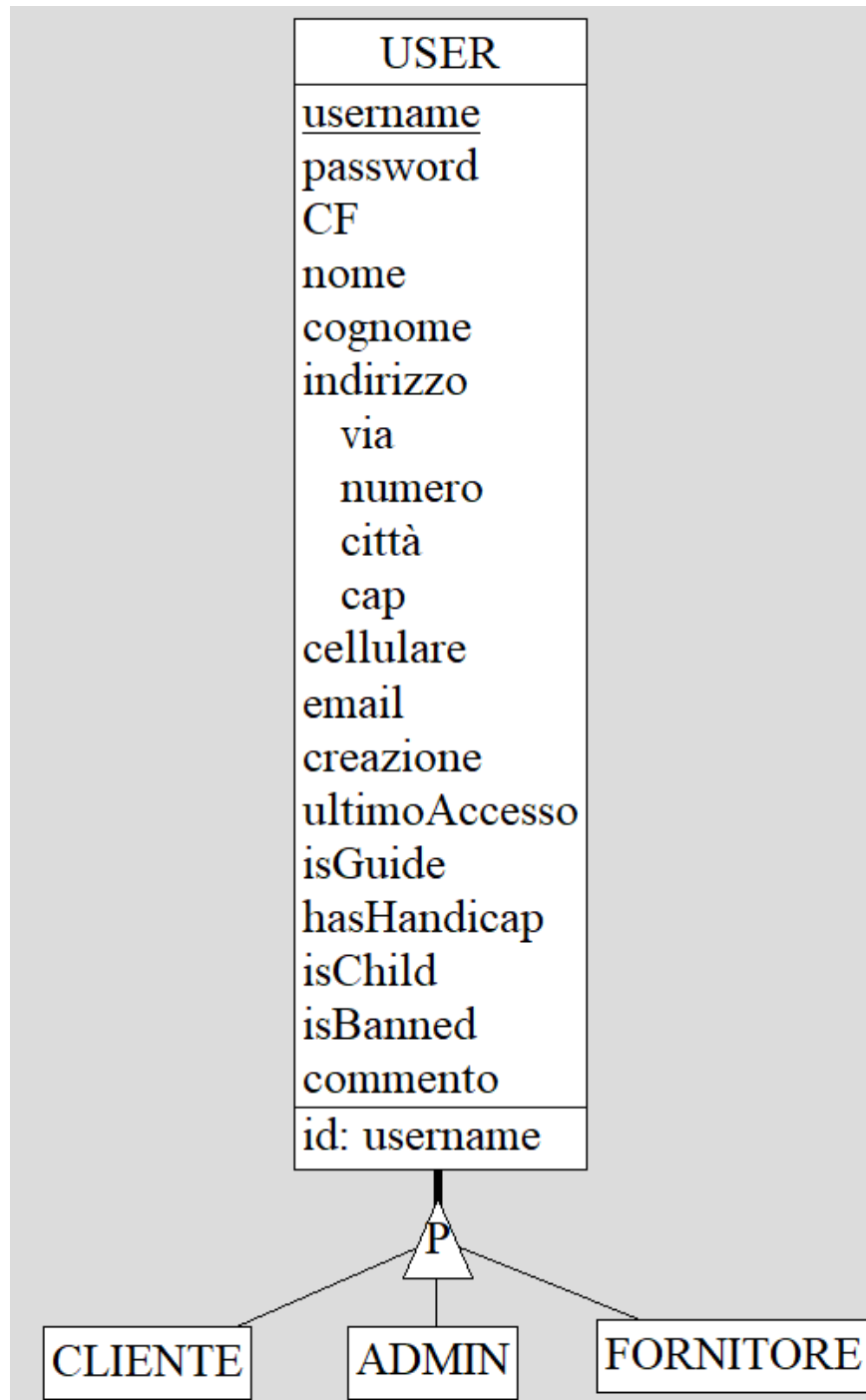
2.3 Schema Iniziale

Qui viene riportato uno schema provvisorio creato dopo una prima analisi dei requisiti e dei concetti fondamentali. Nei paragrafi successivi le entità e le relazioni verranno raffinate e lo schema verrà corretto.



2.4 Schema scheletro prima delle correzioni

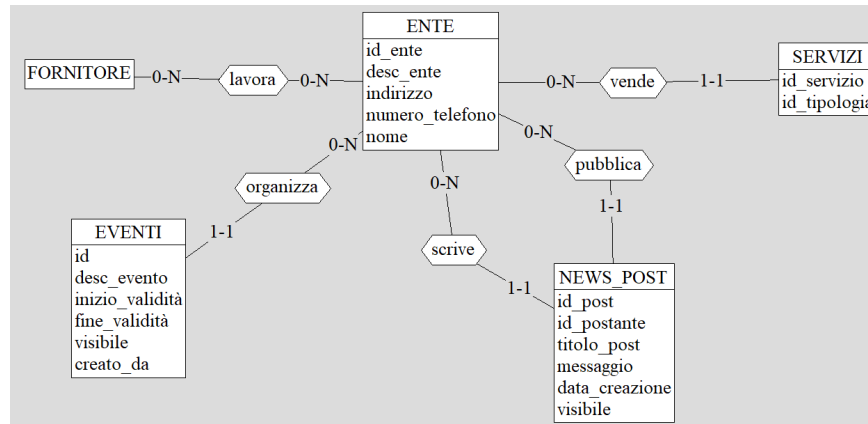
2.4.1 User



I tre tipi di account che potranno essere creati, **CLIENTE**, **FORNITORE**,

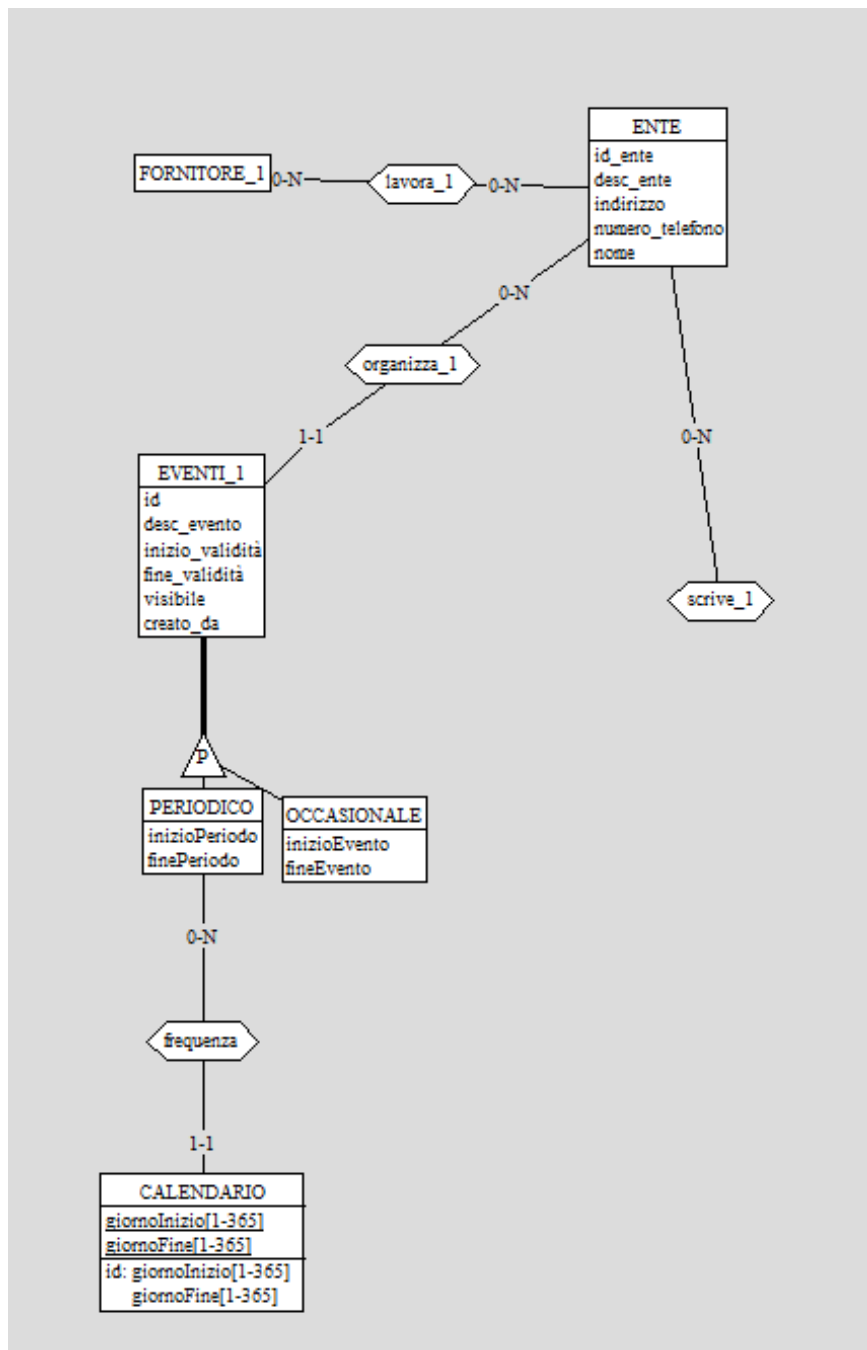
ADMIN, saranno sottocategorie di **USER**. Data la possibilità che una persona crei più di un account, ad esempio un fornitore vuole anche essere un cliente, l'identificazione avverrà tramite **username** e non il **CF**(codice fiscale).

2.4.2 Ente



L'ente è l'organizzazione che vende i servizi e organizza gli eventi. I fornitori possono lavorare per più di un ente. Gli enti creano post e decidono quando pubblicarli/renderli pubblici.

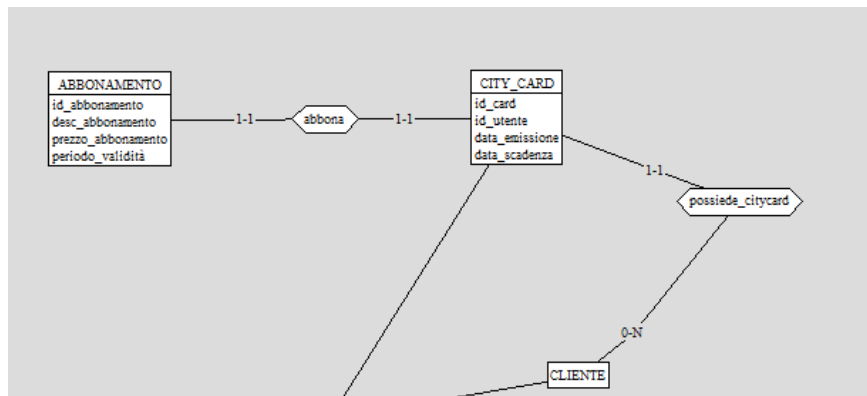
2.4.3 Eventi



I servizi sono gli avvenimenti gratuito accessibili con o senza prenotazione. Sono organizzati dagli enti, un ente può organizzare più eventi ma un evento appar-

tiene ad un solo ente. Un evento può accadere in modalità periodica (ad esempio eventi a tema natalizio) oppure in modo occasionale (ad esempio inaugurazione di una nuova gelateria in centro)

2.4.4 CityCard



Ogni cliente può ottenere una CityCard che permetterà l'utilizzo gratuito dei mezzi pubblici e la prenotazione e l'acquisto di servizi forniti dagli enti comunali e regionali. Per attivare la CityCard sarà necessario sottoscrivere ad un abbonamento a scelta tra tre diverse durate, dalla scelta fatta varierà la validità e gli sconti offerti per l'acquisto dei servizi.

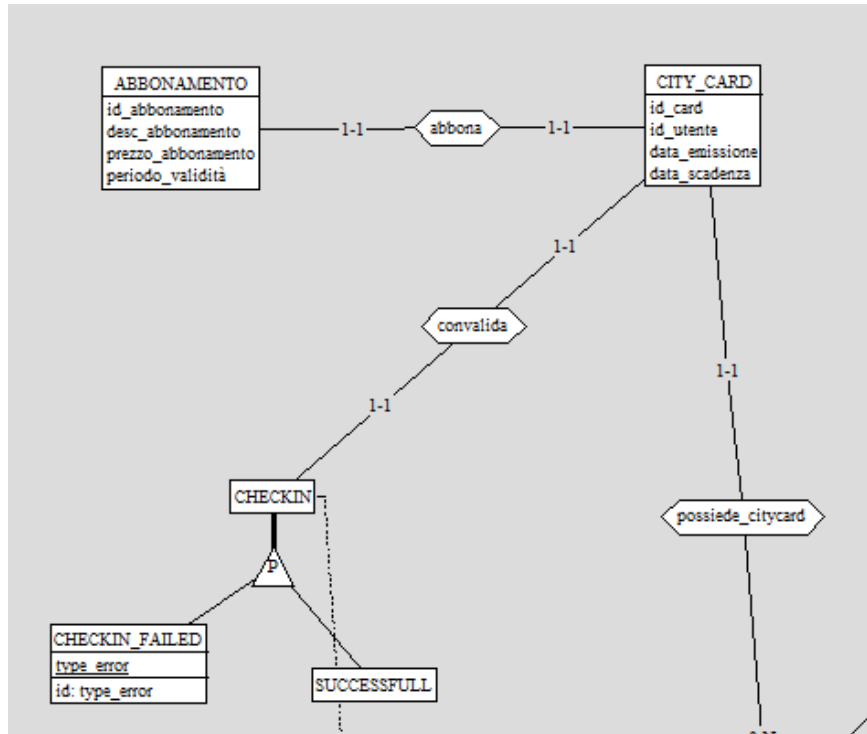
2.4.5 Carta di credito

Ogni cliente, tramite la carta di credito, potrà acquistare la city card e pagare gli enti che forniscono i servizi

2.4.6 Sezione delle news

Una volta avuto accesso al portale ci sarà una sezione dedicata alle notizie relative ad eventi o servizi. L'ente potrà creare un post in cui pubblicizza i servizi offerti o un'attività l'evento che sta organizzando

2.4.7 Abbonamento



L'abbonamento è lo strumento che permette l'attivazione della city card e, di conseguenza, l'acquisto di servizi. Ogni abbonamento ha una sua tipologia che l'utente potrà scegliere di acquistare. A prescindere dalla tipologia ogni abbonamento ha un suo prezzo e un periodo di validità.

2.4.8 Sconti

Lo sconto viene applicato in base ad alcune variabili:

- età del cliente (se è sopra ai 65 anni di età)
- se si è formato un gruppo si avrà una riduzione del servizio

Lo sconto si caratterizza per la percentuale di riduzione.

3 Progettazione Logica

3.1 Stima del volume dei dati

Si fornisce in questa fase una tabella contenente il numero medio di istanze di ogni entità e associazione dello schema globale.

Table 3: Stima del volume di dati

Concetto	Costrutto	Volume	Descrizione
Utente	Entità	10 000	Si stima che i clienti, gli enti e gli amministratori siano numerosi
Possiede CityCard	Relazione	9000	I clienti potrebbero registrarsi ma non avere una CityCard attiva
Ente	Entità	500	Fornisce i servizi che il cliente usufruisce
Crea ente	Relazione	500	Ogni ente è creato da un fornitore
Lavora	Relazione	400	Non tutti i fornitori lavorano per un ente
Servizi	Entità	1500	Data la cardinalità mi aspetto una stima di 3 servizi ad ente
Abbonamento	Entità	9000	La stima è fatta su cardinalità minima 1, ogni cliente possiede un abbonamento
Sottoscrizione	Relazione	9000	La cardinalità minima è 1-1 quindi per ogni CityCard è presente un abbonamento
Carta di credito	Entità	10 000	Si stima che per ogni cliente sia associata una carta di credito
Evento	Entità	1000	Si stima che vengano organizzati almeno 2 eventi per ente
Organizzazione evento	Relazione	1000	Essendo cardinalità 1-1 per ogni evento è prevista un'organizzazione
Partecipazione	Relazione	900	Non tutti i clienti possono essere interessati ad uno specifico evento
Recensione	Relazione	1200	Ogni cliente può lasciare una recensione per servizio o evento ma non è detto che tutti i clienti lascino un'opinione
Trasporto pubblico	Entità	2	Sono presenti due tipi di trasporto previsti all'interno dell'abbonamento

3.2 Operazioni richieste e la loro frequenza

In questa fase esponiamo una tabella delle operazioni utilizzate per costituire una stima delle principali operazioni richieste dal sito.

Table 4: Operazioni comuni

Cod	Nome Operazione	Freq	Tipc	Descrizione
0.1	Accedere tramite login	50	I	Login dell'account
0.2	Registrarsi alla piattaforma	100	I	x Registrazione al sito

Table 5: Operazioni richieste da cliente

Cod	Nome Operazione	Freq	Tipc	Descrizione
1	Acquistare un servizio	100	I	Il cliente può acquistare un servizio
2	Lasciare una recensione riguardo un servizio acquistato	70	I	Il cliente rilascia una votazione
3	Consultare la lista degli acquisti fatti	80	B	Consulta una lista
4	Prenotare un evento	100	I	Chiede all'applicativo una disponibilità di eventi

Table 6: Operazioni richieste fornitore

Cod	Nome Operazione	Freq	Tipc	Descrizione
5	Creare servizi	5/giorno	I	Il fornitore va a creare servizi
6	Creare eventi	5	I	Il fornitore crea eventi
7	Consultare statistiche	50	B	Il fornitore va a consultare una tabella contenente i dati
8	Consultare il saldo	50	B	Il fornitore può consultare il saldo degli acquisti fatti dai clienti
x	x	x	x	x

Table 7: Operazioni richieste amministratore

Cod	Nome Operazione	Freq	Tipc	Descrizione
9	Bannare tutti i tipi di account	50	I	L'admin può interdire l'accesso alla piattaforma
10	Reset delle recensioni 630	I	Può cancellare le recensioni	
11	Consultare la lista degli enti	50	B	Può consultare la lista degli enti registrati

3.3 Frequenza e costo degli accessi

3.4 Analisi delle ridondanze

In questa fase ci occuperemo dell'analisi delle ridondanze cioè quella informazioni che possono essere ricavate altrove. Una ridondanza quindi corrisponde ad un dato che può essere derivato, cioè ottenuto attraverso una serie di operazioni, da altri dati. Se si mantiene la ridondanza si semplificano alcune interrogazioni ma si occupa maggior spazio. All'interno del nostro schema possiamo fare alcune considerazioni.

- L'attributo `numero_partecipanti` è derivabile da una operazione di conteggio delle istanze di clienti che hanno prenotato un evento. Dobbiamo tenere conto delle frequenze di esecuzione.
 - operazione 1: si tiene il conteggio ogni volta che un cliente prenota un evento
 - operazione 2: vengono visualizzati tutti i dati di un evento

3.5 Traduzioni delle operazioni in Query SQL

In questa sezione tradurremo tutte le singole operazioni elementari che abbiamo finora analizzato solo attraverso gli schemi di navigazione dell'E/R in vere e proprie query in linguaggio SQL. Da notare che per facilitare lo dell'applicazione web alcune query sono multiple e di conseguenza è stata posta l'opzione `multipleStatements`.

3.5.1 Recuperare tutti gli utenti

```
SELECT id_user, nome, cognome, email, bannato, descrizione_ruolo
  ↪ AS ruolo
FROM users u
```

```
JOIN ruoli r
ON u.id_ruolo = r.id_ruolo;
```

3.5.2 Recuperare tutti gli enti con e senza recensioni

Seleziona tutti gli enti con e senza recensione, COALESCE restituisce 0 se non c'è una media, i due JOIN imitano una FULL JOIN .

```
SELECT e.id_ente, e.nome AS nome_ente, descrizione, e.indirizzo
      ↳ AS indirizzo_ente, numero_telefono, e.id_user, u.nome AS
      ↳ nome_user, cognome, email, cf, telefono, id_ruolo,
      ↳ media_recensioni
FROM enti e
JOIN users u
ON e.id_user = u.id_user
LEFT JOIN (
SELECT s.id_ente AS id_ente, avg(coalesce(votazione, 0)) AS
      ↳ media_recensioni
FROM recensioni r
JOIN servizi s
ON s.id_servizio = r.id_servizio
GROUP BY s.id_ente
) AS r
ON r.id_ente = e.id_ente
UNION
SELECT e.id_ente, e.nome AS nome_ente, descrizione, e.indirizzo
      ↳ AS indirizzo_ente, numero_telefono, e.id_user, u.nome AS
      ↳ nome_user, cognome, email, cf, telefono, id_ruolo,
      ↳ media_recensioni
FROM enti e
JOIN users u
ON e.id_user = u.id_user
RIGHT JOIN (
SELECT s.id_ente AS id_ente, avg(coalesce(votazione, 0)) AS
      ↳ media_recensioni
FROM recensioni r
JOIN servizi s
ON s.id_servizio = r.id_servizio
GROUP BY s.id_ente
) AS r
ON r.id_ente = e.id_ente
```

3.5.3 Recupero lista eventi

```

SELECT ev.*, en.nome AS organizzatore, periodi.*
FROM eventi ev
JOIN enti en
ON ev.id_ente = en.id_ente
JOIN periodi
ON periodi.id_periodo = ev.id_periodo
WHERE now() < ev.fine_validita
ORDER BY ev.inizio_validita DESC;

```

3.5.4 Recupero lista servizi

```

SET @id_user = ?;
SET @cityCard = ( SELECT id_city_card
FROM city_card
WHERE id_user = @id_user and data_scadenza > now());
SET @percentToPay = (SELECT (100 - sconti.percentuale_sconto)
    ↪ / 100
FROM sconti
JOIN listino_abbonamenti ls
ON sconti.id_sconto = ls.id_sconto
JOIN sottoscrizioni_abbonamento sa
ON ls.id_sconto = sa.id_listino_abbonamento
WHERE sa.id_city_card = @cityCard);
SELECT se.*, en.nome AS organizzatore, se.prezzo_servizio *
    ↪ @percentToPay AS prezzo_scontato
FROM servizi se
JOIN enti en
ON se.id_ente = en.id_ente
WHERE now() < se.fine_validita;

```

3.5.5 Recupero acquisti di un utente

```

SELECT sa.data_acquisto, sa.prezzo_pagato, cc.id_city_card, sa.
    ↪ num_carta_credito, servizi.descrizione_servizio AS
    ↪ nome_servizio, u.id_user, sa.id_servizio
FROM servizi_acquistati sa
JOIN city_card cc
ON cc.id_city_card = sa.id_city_card
JOIN users u
ON u.id_user = cc.id_user
JOIN servizi
ON servizi.id_servizio = sa.id_servizio
WHERE cc.data_scadenza > now() and cc.id_user = ?

```

```
ORDER BY sa.data_acquisto DESC;
```

3.5.6 Reitorna statistiche amministratore

Uso le variabili per rendere più leggibili le query e le unisco in un unico SELECT per rendere l'inserimento in tabello più semplice.

```
INSERT INTO users (username, nome, cognome, email,
password, id_ruolo, data_creazione)
VALUES (SET @numeroCheckin = (SELECT count(coalesce(
    ↪ id_check,0)) FROM checks);
SET @numeroCheckinFalliti = (SELECT count(coalesce(id_check
    ↪ ,0)) FROM checks WHERE id_check != 1);
SET @numeroCityCardAttive = (SELECT count(coalesce(
    ↪ id_city_card,0)) FROM city_card WHERE data_scadenza >
    ↪ now());
SET @numeroEventiAttivi = (SELECT count(coalesce(id_evento
    ↪ ,0)) FROM eventi WHERE fine_validita > now());
SET @numeroServiziAttivi = (SELECT count(coalesce(id_servizio
    ↪ ,0)) FROM servizi WHERE fine_validita > now());

SELECT
@numeroCheckin AS numero_checkin,
@numeroCheckinFalliti AS numeroCheckinFalliti,
@numeroCityCardAttive AS numeroCityCardAttive,
@numeroEventiAttivi AS numeroEventiAttivi,
@numeroServiziAttivi AS numeroServiziAttivi);
```

3.5.7 Recupero statistiche fornitore

```
SET @id_utente = 2;
SET @id_ente = ( SELECT c.id_ente
FROM collaborazioni c
WHERE c.id_user = @id_utente and c.fine_collaborazione is null
    ↪ limit 1);

SET @saldo = (SELECT saldo FROM enti WHERE id_ente =
    ↪ @id_ente);
SET @numeroEventiAttivi = (SELECT count(coalesce(id_evento
    ↪ ,0)) FROM eventi WHERE fine_validita > now() and
    ↪ id_ente = @id_ente);
SET @numeroServiziAttivi = (SELECT count(coalesce(id_servizio
    ↪ ,0)) FROM servizi WHERE fine_validita > now() and
    ↪ id_ente = @id_ente);
```



```

SELECT
IFNULL(@saldo,0) AS saldo,
@numeroEventiAttivi AS numeroEventiAttivi,
@numeroServiziAttivi AS numeroServiziAttivi;

```

3.5.8 Recupero listino abbonamenti

```

SELECT l.*, s.percentuale_sconto
FROM listino_abbonamenti l
JOIN sconti s
ON l.id_sconto = s.id_sconto
WHERE l.data_disattivazione is null or l.data_disattivazione < now
    ↪ ();

```

3.5.9 Creazione nuovo ente

```

INSERT INTO enti (nome, descrizione, indirizzo, numero_telefono,
    ↪ id_user)
VALUES (?, ?, ?, ?, ?);

```

3.5.10 Inserimento voto utente

Dato che un utente può avere una recensione per servizio cancello la inserisco il voto nuovo, calcolo la media, la inserisco nel servizio.

```

SET @idUser = ?;
SET @idServizio = ?;

DELETE
FROM recensioni
WHERE id_user = @idUser and id_servizio = @idServizio;

insert into recensioni (id_user, votazione, id_servizio) VALUES(
    ↪ @idUser,?,@idServizio);

SET @media = (SELECT AVG(votazione)
FROM recensioni
WHERE id_user = @idUser and id_servizio = @idServizio);

UPDATE servizi
SET media_recensioni = @media
WHERE id_servizio = @idServizio;

```

3.5.11 Reset recensioni di un ente

Reset di tutte le recensioni riguardanti un ente

```
SET @id_ente = ?;

DELETE recensioni
FROM recensioni
JOIN servizi s
ON s.id_servizio = recensioni.id_servizio
WHERE s.id_ente = @id_ente;

UPDATE servizi s
SET media_recensioni = null
WHERE s.id_ente = @id_ente;
```

3.5.12 Creazione servizio

Crea un servizio, "LIMIT 1" non dovrebbe servire in quanto un fornitore può essere associato con solo un ente ma protegge da possibili errori come una doppia scrittura dello stesso record

```
INSERT INTO servizi (descrizione_servizio, indirizzo_servizio,
    ↳ id_ente, prezzo_servizio)
VALUES (?, ?,
    ( SELECT c.id_ente
    FROM collaborazioni c
    WHERE c.id_user = ? and c.fine_collaborazione is null LIMIT 1),
    ↳ ?);
```

3.5.13 Registrazione nuovo utente

```
INSERT INTO users (username, nome, cognome, email, password,
    ↳ id_ruolo, data_creazione)
VALUES (?, ?, ?, ?, ?, ?, ?);
```

3.5.14 Registrazione nuova carta di credito

```
INSERT INTO carte_credito (num_carta_credito, cognome_associato
    ↳ , nome_associato, mese_scadenza, anno_scadenza, id_user)
VALUES (?, ?, ?, ?, ?, ?);
```

3.5.15 Creazione nuova CityCard

```
INSERT INTO city_card (id_user)
VALUES('?');
```

3.5.16 Disattivazione CityCard

Per disattivare una card pongo la data di scadenza a now(). Per controllare se la card è attiva controllo se il momento di disattivazione è già passato. In ogni momento un utente può avere al massimo una CityCard attiva, quindi per default disattivo tutte quelle attive.

```
\dfrac{\textbf{update}}{\dfrac{\text{num}}{\text{den}}} city\_card
SET data\_scadenza = now()
WHERE id\_user = ? and data\_scadenza > now();
```

3.5.17 Rendere predefinita una carta di credito

rendo predefinita una carta di credito di un utente. Rendo non predefinite tutte le altre. Query doppia, una delle alternative a quello fatto in ban

```
UPDATEcarte_credito
SET predefinita = 0
WHERE id_user = ?;

UPDATEcarte_credito
SET predefinita = 1
WHERE num_carta_credito = ?;
```

3.5.18 Sottoscrizione abbonamento

```
INSERT INTO sottoscrizioni.abbonamento (id_listino.abbonamento
↪ , id_city_card, num_carta_credito)
VALUES (?,
( SELECT id_city_card
FROM city_card
WHERE id_user = ? and data_scadenza > now()),
( SELECT num_carta_credito
FROM carte_credito
WHERE id_user = ? and predefinita = 1));
```

3.5.19 Associa utente a ente

```
INSERT INTO collaborazioni
(id_user, id_ente)
VALUES (?, ?);
```

3.5.20 Effettua check-in

```
INSERT INTO checks (id_city_card, id_mezzo, id_stato)
VALUES (( SELECT id_city_card
FROM city_card
WHERE id_user = ? and data_scadenza > now()),
?, ?);
```

3.5.21 Ottieni CityCard di un utente

```
SELECT *, (if (data_scadenza > now(), "attiva", "non-attiva")) AS
↪ stato
FROM city_card
WHERE id_user = ?
ORDER BY id_city_card DESC
```

3.5.22 Partecipa a un evento

```
INSERT INTO partecipazioni (id_evento, id_user) VALUES (?,?);
```

3.5.23 Edita Utente

```
UPDATE users
SET username = ?, nome = ?, cognome = ?, email = ?, password =
↪ ?, indirizzo = ?, telefono = ?, cf = ?
WHERE id_user = ?;
```

3.5.24 Banna/sbanna utente

Query tecnicamente sbagliata ma ho provato ad usare il trucco dell'INNER JOIN per aggirare il problema. L'alternativa è fare due query separate ban/unban.

```
UPDATE users u
INNER JOIN users u1
ON u.id_user = u1.id_user
SET u.bannato = CASE
WHEN (
  (
    SELECT u1.bannato
    WHERE u1.id_user = ?
  ) = 1
) THEN 0
ELSE 1
END
WHERE u.id_user = ?;
```

3.5.25 Cancellata carta di credito

```
DELETE FROM carte_credito
WHERE num_carta_credito = ?;
```

3.5.26 Effettua login

```
SELECT u.*, r.descrizione_ruolo AS ruolo
FROM users u
JOIN ruoli r
ON u.id_ruolo = r.id_ruolo
WHERE username = ? AND password = ?;
```

3.5.27 Ritorna, se presente, la CityCard attiva dell'utente

```
SELECT *
FROM city_card
WHERE id_user = ? and data_scadenza > now();
```

3.5.28 Ritorna ente associato all'utente

```
SELECT *
FROM collaborazioni
WHERE id_user = ? and fine_collaborazione is null;
```

3.5.29 ritorna, se presente, la sottoscrizione attiva attiva dell'utente

```
SELECT sa.*  
FROM sottoscrizioni_abbonamento sa  
JOIN city_card cc  
ON sa.id_city_card = cc.id_city_card  
JOIN users u  
ON cc.id_user = u.id_user  
WHERE u.id_user = ?;
```

3.5.30 Ritorna lista check-in fatti da un utente

```
SELECT ch.*, sc.descrizione_stato, m.*  
FROM checks ch  
JOIN city_card cc  
ON ch.id_city_card = cc.id_city_card  
JOIN users u  
ON cc.id_user = u.id_user  
JOIN stati_check sc  
ON ch.id_stato = sc.id_stato  
JOIN mezzi m  
ON ch.id_mezzo = m.id_mezzo  
WHERE u.id_user = ?;
```

3.5.31 Recupera enti

```
SELECT * FROM enti  
WHERE nome like ?;
```

3.5.32 Recupera carte di credito di un utente

```
SELECT c.*, u.nome, u.cognome  
FROM carte_credito c  
JOIN users u  
ON c.id_user = u.id_user  
WHERE c.id_user = ?;
```

3.5.33 Compra servizio

Compra un servizio tenendo conto dello sconto concesso dalla fascia di abbonamento posseduto, poi aggiunge la somma pagata al saldo dell'ente che eroga il servizio. La query è stata fatta con le variabili perché senza incappava nello

stesso problema del ban utente, cioè conflitto in caso di stesso attributi in INSERT E WHERE della stessa query.

```
SET @id_user = ?;
SET @id_servizio = ?;
SET @idEnte = (
SELECT enti.id_ente
FROM enti
JOIN servizi
ON enti.id_ente = servizi.id_ente
WHERE servizi.id_servizio = @id_servizio);
SET @creditCard = ( SELECT num_carta_credito
FROM carte_credito
WHERE id_user = @id_user and predefinita = 1);
SET @cityCard = ( SELECT id_city_card
FROM city_card
WHERE id_user = @id_user and data_scadenza > now());
SET @percentToPay = (SELECT (100 - sconti.percentuale_sconto)
→ / 100
FROM sconti
JOIN listino_abbonamenti ls
ON sconti.id_sconto = ls.id_sconto
JOIN sottoscrizioni_abbonamento sa
ON ls.id_sconto = sa.id_listino_abbonamento
WHERE sa.id_city_card = @cityCard);
SET @paidPrice = ( SELECT prezzo_servizio * @percentToPay
FROM servizi
WHERE id_servizio = @id_servizio);

INSERT INTO servizi_acquistati (id_servizio, prezzo_pagato,
→ num_carta_credito, id_city_card)
VALUES (@id_servizio, @paidPrice, @creditCard,@cityCard );

UPDATE enti
SET saldo = saldo + @paidPrice
WHERE id_ente = @idEnte;
```

3.5.34 Creazione evento periodico

Per ottenere l'ultimo ID inserito in periodi una opzione poteva essere l'utilizzo di LAST_INSERT_ID ma questo richiedeva l'attivazione di PersistentConnections al momento della connessione.

```
SET @idUser = ?;
SET @idEnte = ( SELECT c.id_ente
FROM collaborazioni c
```

```

WHERE c.id_user = @idUser and c.fine_collaborazione is null limit
    ↪ 1);

INSERT INTO periodi (lunedì, martedì, mercoledì, giovedì, venerdì,
    ↪ sabato, domenica)
VALUES (?, ?, ?, ?, ?, ?, ?);

SET @idPeriodo = (SELECT MAX(id_periodo) FROM periodi);

INSERT INTO eventi (id_periodo, nome_evento, num_partecipanti,
    ↪ id_ente)
VALUES (@idPeriodo, ?, ?, @idEnte);

```

3.5.35 Creazione evento occasionale

```

SET @idUser = ?;
SET @idEnte = ( SELECT c.id_ente
FROM collaborazioni c
WHERE c.id_user = @idUser and c.fine_collaborazione is null limit
    ↪ 1);

INSERT INTO eventi (id_periodo, nome_evento, num_partecipanti,
    ↪ id_ente)
VALUES (1, ?, ?, @idEnte);

```

4 Progetto dello schema concettuale

Termini utilizzati negli schemi:

- - Account-Utente
- - Lista di servizi
- - Transazione

Lo schema concettuale si avvarrà delle seguenti entità e associazioni:

Nome	Tipo	Descrizione
Account	E	
Transazione	R	

Table 8: Tavolo originale

Nome Account Transazione	Tipo E R	Descrizione

Table 9: Tavolo di prova