



Basi di Dati  
Progetto A.A. 2023/2024

# SISTEMA DI ASTE ONLINE

0267710  
Alessio Torrioni

## Indice

1. Descrizione del Minimondo .....	2
2. Analisi dei Requisiti .....	4
3. Progettazione concettuale .....	8
4. Progettazione logica .....	13
5. Progettazione fisica .....	22

## 1. Descrizione del Minimondo

Una casa d'aste intende realizzare un sistema online di aste. Il sistema deve consentire agli amministratori la gestione degli oggetti che si vogliono pubblicare e tutto il ciclo di vita delle aste. Gli utenti del sistema, previa registrazione, hanno la possibilità di fare offerte su un qualsiasi oggetto. Al termine dell'asta, l'offerta maggiore sarà quella che avrà vinto l'asta. Alla registrazione, gli utenti devono comunicare il codice fiscale, il nome, il cognome, la data di nascita, la città di nascita, le informazioni sulla propria carta di credito (intestatario, numero, data di scadenza, codice CVV). Inoltre, essi devono fornire un indirizzo cui consegnare eventuali oggetti acquistati.

Gli amministratori gestiscono l'inserimento degli oggetti. Ogni oggetto è caratterizzato da un codice alfanumerico univoco, da una descrizione, da uno stato (ad esempio "come nuovo", "in buone condizioni", "non funzionante", ecc.), da un prezzo di base d'asta, e da una descrizione delle dimensioni. Quando viene inserito un nuovo oggetto nel sistema, gli amministratori possono decidere la durata dell'asta, da un minimo di un giorno ad un massimo di sette giorni. Inoltre, a ciascuna asta viene associata una categoria. Le categorie appartengono ad un titolario gerarchico, organizzato su un massimo di tre livelli. La gestione delle categorie degli oggetti afferisce sempre agli amministratori del sistema.

Gli utenti del sistema possono visualizzare in qualsiasi momento tutte le aste aperte. Quando un'asta viene visualizzata, gli utenti ottengono tutte le informazioni legate allo stato attuale della stessa, tra cui il tempo mancante alla chiusura, il numero di offerte fatte, l'importo dell'offerta massima attuale. Non possono però visualizzare chi è che ha effettuato l'offerta massima.

Dato un oggetto in asta, gli utenti possono fare un'offerta, maggiore del valore attuale di offerta. La granularità di incremento delle offerte è di multipli di 50 centesimi di euro. Inoltre, un utente che ha attualmente piazzato l'offerta massima, può sfruttare la funzionalità di "controfferta automatica". Tale funzionalità permette all'utente di indicare un importo massimo con cui si intende rilanciare l'offerta, qualora un altro utente faccia un'offerta maggiore. La gestione delle offerte pertanto funziona nel modo seguente. L'utente A indica un importo I con cui vuole rilanciare l'offerta nei confronti dell'utente B che è attualmente il migliore offerente. L'utente B ha anche indicato un importo di

33 controfferta C. Se  $C > I$ , il sistema indicherà come miglior offerente l'utente A, con  
34 importo temporaneo I, ma immediatamente dopo indicherà nuovamente l'utente B come  
35 migliore offerente, con un importo di  $I + 0,50\text{€}$ .

36  
37 Il sistema tiene traccia, per ogni oggetto, di tutte le offerte che sono state fatte e  
38 dell'istante temporale in cui queste sono state inserite nel sistema. Ciò significa che tutte  
39 le transazioni automatiche generate dal sistema di controfferta automatica devono essere  
40 registrate nel sistema.

41  
42 Gli utenti, in ogni momento, possono visualizzare l'elenco degli oggetti aggiudicati e  
43 l'elenco degli oggetti per i quali è presente un'asta in corso cui hanno fatto almeno  
44 un'offerta.

## 2. Analisi dei Requisiti

### Identificazione dei termini ambigui e correzioni possibili

Linea	Termine	Nuovo termine	Motivo correzione
2	Pubblicare	Inserire	Stesso concetto, manteniamo una coerenza lessicale con i termini successivi
4	Oggetto	Oggetto in asta	Un utente può effettuare offerte su un oggetto su cui è ancora aperta un'asta, non su un qualsiasi oggetto
15	Asta	Oggetto	Una categoria viene assegnata ad un oggetto, non ad un'asta
16	Titolario	Insieme di titoli	Termine che esprime in modo più semplice e comprensibile il concetto
39	Transazione	Offerta	Il sistema di controfferta automatica effettua un'offerta nell'asta, non una transazione

### Specifica disambiguata

Una casa d'aste intende realizzare un sistema online di aste. Il sistema deve consentire agli amministratori la gestione degli oggetti che si vogliono inserire e tutto il ciclo di vita delle aste. Gli utenti del sistema, previa registrazione, hanno la possibilità di fare offerte su un qualsiasi oggetto in asta. Al termine dell'asta, l'offerta maggiore sarà quella che avrà vinto l'asta. Alla registrazione, gli utenti devono comunicare il codice fiscale, il nome, il cognome, la data di nascita, la città di nascita, le informazioni sulla propria carta di credito (intestatario, numero, data di scadenza, codice CVV). Inoltre, essi devono fornire un indirizzo cui consegnare eventuali oggetti acquistati.

Gli amministratori gestiscono l'inserimento degli oggetti. Ogni oggetto è caratterizzato da un codice alfanumerico univoco, da una descrizione, da uno stato (ad esempio "come nuovo", "in buone condizioni", "non funzionante", ecc.), da un prezzo di base d'asta, e da una descrizione delle dimensioni. Quando viene inserito un nuovo oggetto nel sistema, gli amministratori possono decidere la durata dell'asta, da un minimo di un giorno ad un massimo di sette giorni. Inoltre, a ciascuna oggetto viene associata una categoria. Le categorie appartengono ad un insieme di titoli gerarchico, organizzato su un massimo di tre livelli. La gestione delle categorie degli oggetti afferisce sempre agli amministratori del sistema.

Gli utenti del sistema possono visualizzare in qualsiasi momento tutte le aste aperte. Quando un'asta viene visualizzata, gli utenti ottengono tutte le informazioni legate allo stato attuale della stessa, tra cui il tempo mancante alla chiusura, il numero di offerte fatte, l'importo dell'offerta massima attuale. Non possono però visualizzare chi è che ha effettuato l'offerta massima.

Dato un oggetto in asta, gli utenti possono fare un'offerta, maggiore del valore attuale di offerta. La granularità di incremento delle offerte è di multipli di 50 centesimi di euro. Inoltre, un utente che ha attualmente piazzato l'offerta massima, può sfruttare la funzionalità di "controfferta automatica". Tale funzionalità permette all'utente di indicare un importo massimo con cui si intende rilanciare l'offerta, qualora un altro utente faccia un'offerta maggiore. La gestione delle offerte pertanto funziona nel modo seguente. L'utente A indica un importo I con cui vuole rilanciare l'offerta nei confronti dell'utente B che è attualmente il migliore offerente. L'utente B ha anche indicato un importo di controfferta C. Se  $C > I$ , il sistema indicherà come miglior offerente l'utente A, con importo temporaneo I, ma immediatamente dopo indicherà nuovamente l'utente B come migliore offerente, con un importo di  $I + 0,50\text{€}$ .

Il sistema tiene traccia, per ogni oggetto, di tutte le offerte che sono state fatte e dell'istante temporale in cui queste sono state inserite nel sistema. Ciò significa che tutte le offerte automatiche generate dal sistema di controfferta automatica devono essere registrate nel sistema.

Gli utenti, in ogni momento, possono visualizzare l'elenco degli oggetti aggiudicati e l'elenco degli oggetti per i quali è presente un'asta in corso cui hanno fatto almeno un'offerta.

## Glossario dei Termini

Termine	Descrizione	Sinonimi	Collegamenti
Utente	Persona che può registrarsi all'interno del sistema e può effettuare offerte su degli oggetti in asta per acquistarli.		Offerta, Oggetto in asta, Controfferta
Oggetto in asta	Oggetto inserito nel sistema su cui si possono effettuare offerte		Utente, Categoria, Offerta, Controfferta, Amministratore
Offerta	Offerta, con un importo, relativo ad un oggetto in asta effettuata da un utente.		Utente, Oggetto in asta
Categoria	Classificazione degli oggetti inseriti nel sistema, organizzate in un insieme di titoli gerarchico organizzato su tre livelli		Oggetto in asta

Controfferta	Offerta automatica effettuata dal sistema su un oggetto previa impostazione dell'utente		Utente, Oggetto in asta
Amministratore	Persona che può gestire l'inserimento di oggetti e il ciclo di vita delle rispettive aste		Oggetto in asta

### Raggruppamento dei requisiti in insiemi omogenei

#### Frasi relative a Utente

Gli utenti del sistema, previa registrazione, hanno la possibilità di fare offerte su un qualsiasi oggetto in asta ... Alla registrazione, gli utenti devono comunicare il codice fiscale, il nome, il cognome, la data di nascita, la città di nascita, le informazioni sulla propria carta di credito (intestatario, numero, data di scadenza, codice CVV). Inoltre, essi devono fornire un indirizzo cui consegnare eventuali oggetti acquistati...Gli utenti del sistema possono visualizzare in qualsiasi momento tutte le aste aperte. Quando un'asta viene visualizzata, gli utenti ottengono tutte le informazioni legate allo stato attuale della stessa, tra cui il tempo mancante alla chiusura, il numero di offerte fatte, l'importo dell'offerta massima attuale. Non possono però visualizzare chi è che ha effettuato l'offerta massima. Dato un oggetto in asta, gli utenti possono fare un'offerta, maggiore del valore attuale di offerta ... Gli utenti, in ogni momento, possono visualizzare l'elenco degli oggetti aggiudicati e l'elenco degli oggetti per i quali è presente un'asta in corso cui hanno fatto almeno un'offerta

#### Frasi relative a Oggetto in asta

Ogni oggetto è caratterizzato da un codice alfanumerico univoco, da una descrizione, da uno stato (ad esempio "come nuovo", "in buone condizioni", "non funzionante", ecc.), da un prezzo di base d'asta, e da una descrizione delle dimensioni. Quando viene inserito un nuovo oggetto nel sistema, gli amministratori possono decidere la durata dell'asta, da un minimo di un giorno ad un massimo di sette giorni ... A ciascun oggetto viene associato una categoria...Il sistema tiene traccia, per ogni oggetto, di tutte le offerte che sono state fatte e dell'istante temporale in cui queste sono state inserite nel sistema ... Gli utenti, in ogni momento, possono visualizzare l'elenco degli oggetti aggiudicati.

#### Frasi relative a Offerta

Gli utenti del sistema hanno la possibilità di fare offerte su un qualsiasi oggetto in asta. Al termine dell'asta, l'offerta maggiore sarà quella che avrà vinto l'asta ... Dato un oggetto in asta, gli utenti possono fare un'offerta, maggiore del valore attuale di offerta. La granularità di incremento delle offerte è di multipli di 50 centesimi di euro. Inoltre, un utente che ha attualmente piazzato l'offerta massima, può sfruttare la funzionalità di "controfferta automatica". Tale funzionalità permette all'utente di indicare un importo massimo con cui si intende rilanciare l'offerta, qualora un altro utente faccia un'offerta maggiore. La gestione delle offerte pertanto funziona nel modo seguente. L'utente A indica un importo I con cui vuole rilanciare l'offerta nei confronti dell'utente B che è

attualmente il migliore offerente. L'utente B ha anche indicato un importo di controfferta C. Se  $C > I$ , il sistema indicherà come miglior offerente l'utente A, con importo temporaneo I, ma immediatamente dopo indicherà nuovamente l'utente B come migliore offerente, con un importo di  $I + 0,50\text{€}$  ... Il sistema tiene traccia, per ogni oggetto, di tutte le offerte che sono state fatte e dell'istante temporale in cui queste sono state inserite nel sistema. Ciò significa che tutte le offerte automatiche generate dal sistema di controfferta automatica devono essere registrate nel sistema. Gli utenti, in ogni momento, possono visualizzare l'elenco degli oggetti aggiudicati e l'elenco degli oggetti per i quali è presente un'asta in corso cui hanno fatto almeno un'offerta.

#### **Frasi relative a Categoria**

Le categorie appartengono ad un insieme di titoli gerarchico, organizzato su un massimo di tre livelli. La gestione delle categorie degli oggetti afferisce sempre agli amministratori del sistema.

#### **Frasi relative a Controfferta**

Inoltre, un utente che ha attualmente piazzato l'offerta massima, può sfruttare la funzionalità di "controfferta automatica". Tale funzionalità permette all'utente di indicare un importo massimo con cui si intende rilanciare l'offerta, qualora un altro utente faccia un'offerta maggiore. La gestione delle offerte pertanto funziona nel modo seguente. L'utente A indica un importo I con cui vuole rilanciare l'offerta nei confronti dell'utente B che è attualmente il migliore offerente. L'utente B ha anche indicato un importo di controfferta C. Se  $C > I$ , il sistema indicherà come miglior offerente l'utente A, con importo temporaneo I, ma immediatamente dopo indicherà nuovamente l'utente B come migliore offerente, con un importo di  $I + 0,50\text{€}$ .

Tutte le offerte automatiche generate dal sistema di controfferta automatica devono essere registrate nel sistema.

#### **Frasi relative a Amministratore**

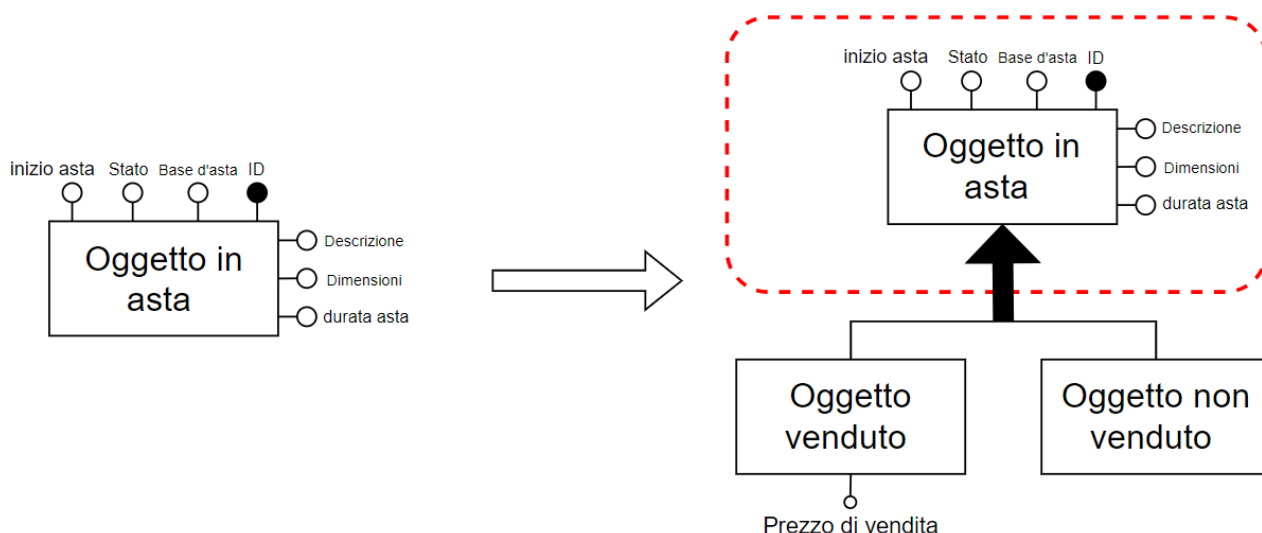
Il sistema deve consentire agli amministratori la gestione degli oggetti che si vogliono inserire e tutto il ciclo di vita delle aste ... Gli amministratori gestiscono l'inserimento degli oggetti. Ogni oggetto è caratterizzato da un codice alfanumerico univoco, da una descrizione, da uno stato (ad esempio "come nuovo", "in buone condizioni", "non funzionante", ecc.), da un prezzo di base d'asta, e da una descrizione delle dimensioni. Quando viene inserito un nuovo oggetto nel sistema, gli amministratori possono decidere la durata dell'asta, da un minimo di un giorno ad un massimo di sette giorni... La gestione delle categorie degli oggetti afferisce sempre agli amministratori del sistema.

### 3. Progettazione concettuale

#### Costruzione dello schema E-R

Nella progettazione concettuale per il sistema di aste online, adottiamo un approccio “inside-out”. Questo approccio si concentra prima sul concetto centrale del sistema, l’Oggetto in asta, e successivamente espande la visione includendo altri concetti correlati. Ogni oggetto in asta è unico e viene identificato da un codice alfanumerico, una descrizione, uno stato, una descrizione delle dimensioni dell’oggetto, una base d’asta, l’inizio e la durata dell’asta. Inizialmente come vediamo nella Figura 1 utilizziamo la generalizzazione per catturare il concetto di oggetto venduto ad un determinato prezzo e di oggetto non venduto, ovvero quando non viene effettuata alcuna offerta allo scadere del tempo prefissato nell’attributo durata asta.

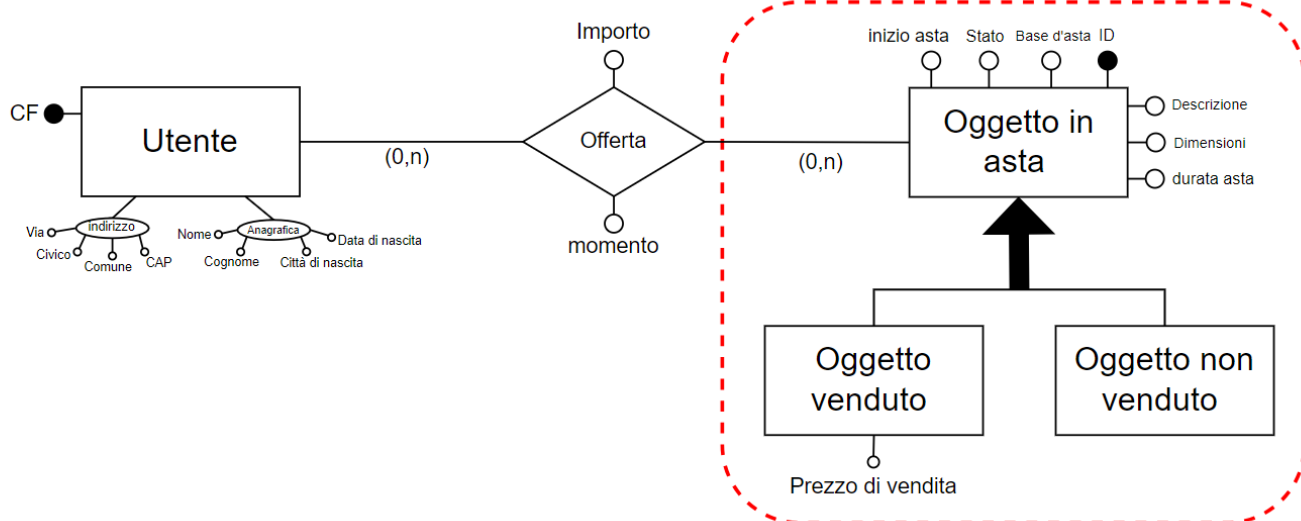
FIGURA 1



Successivamente [Figura 2] viene introdotto il concetto di Utente, coloro che effettuano offerte sugli oggetti in asta. Ogni utente è unico e viene identificato dal codice fiscale, gli attributi anagrafici come nome, cognome, data e città di nascita vengono raggruppati in un attributo multivalore chiamato appunto “anagrafica”. Allo stesso modo gli attributi che riguardano i dati riguardanti l’indirizzo di consegna: via, numero civico, comune e CAP sono raggruppati in un altro attributo multivalore chiamato “indirizzo”. Nell’associazione “offerta” tra utente e oggetto in asta aggiungiamo gli attributi importo e momento dell’offerta in questione.

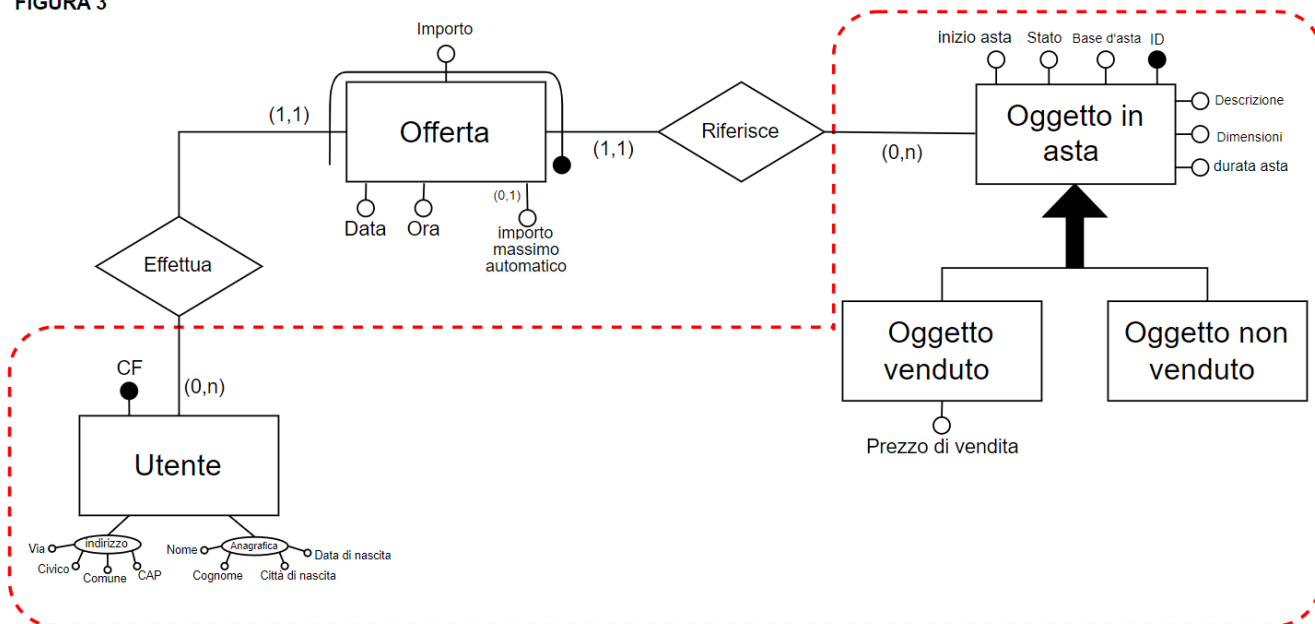


FIGURA 2



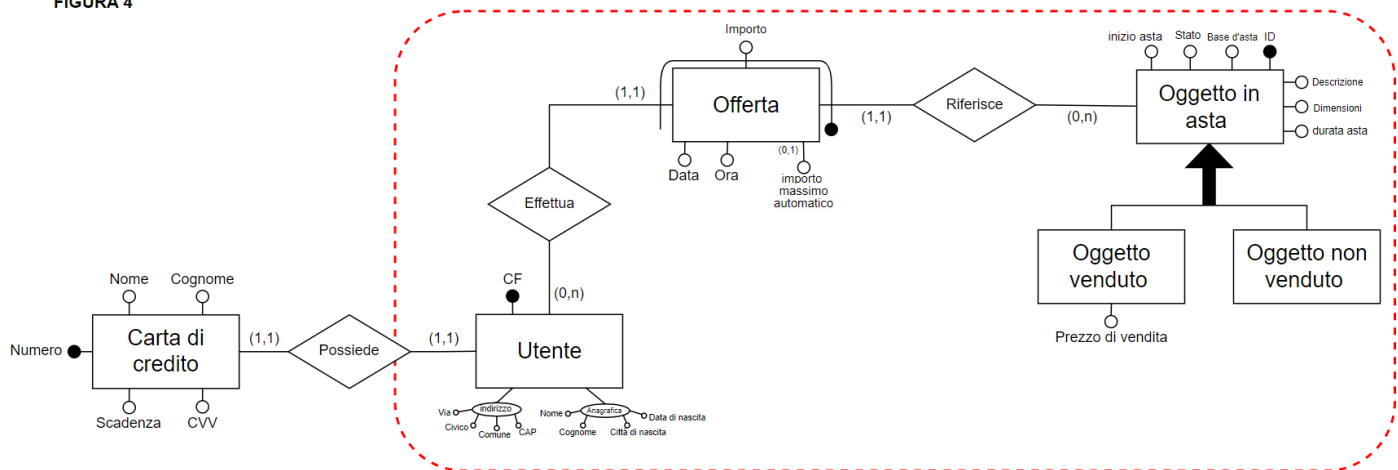
Osserviamo ora che la soluzione precedente riguardo la relazione "Offerta" è valida solo se ogni utente può effettuare una sola offerta su un determinato oggetto in asta perché, per definizione, una occorrenza della relazione Offerta è un insieme di coppie Utente-Oggetto in asta, senza duplicati. Risulta quindi opportuno reificare la relazione rappresentandola come entità. In questo caso, l'identificazione di un'Offerta avviene attraverso l'Utente, l'Oggetto e l'importo dell'Offerta. All' entità Offerta viene aggiunto inoltre un attributo opzionale "importo massimo automatico" che permette di catturare il concetto riguardante la possibilità di un utente di impostare un'offerta automatica fino al raggiungimento di un limite di importo prefissato [ Figura 3 ].

FIGURA 3



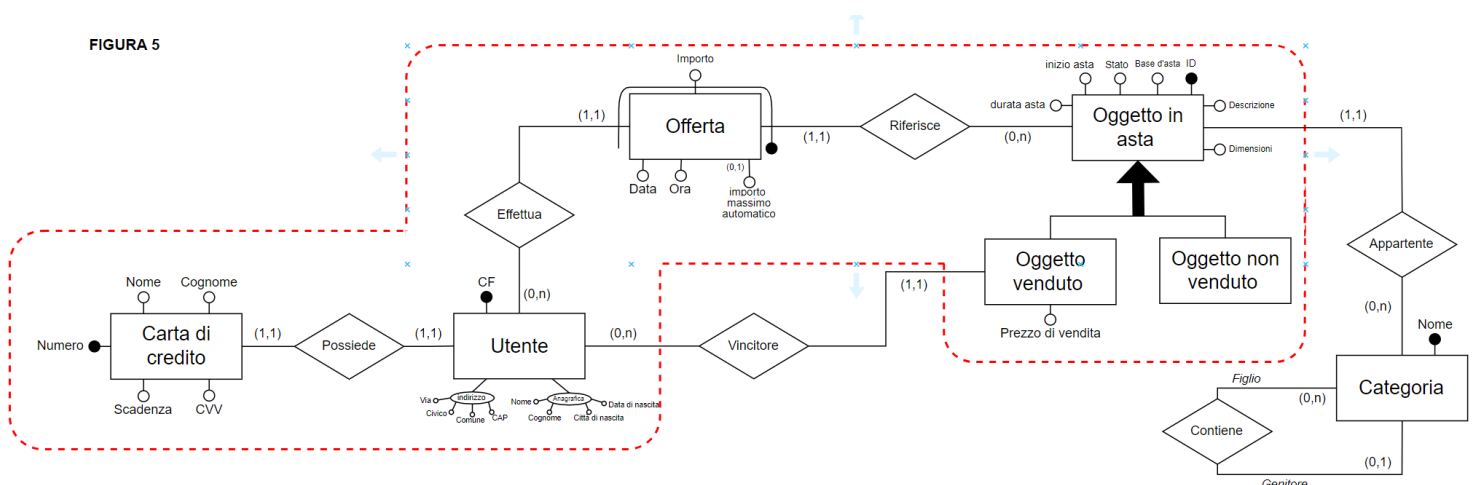
Nella Figura 4 , per mantenere i concetti di carta di credito ed utente distinti, viene aggiunta un'entità apposita per rappresentare i dati della carta di credito, essa viene identificata tramite il numero della carta, come attributi ci sono Nome e Cognome dell'intestatario, il CVV e la data di scadenza. Tale entità viene associata all'entità Utente tramite la relazione "possiede".

FIGURA 4

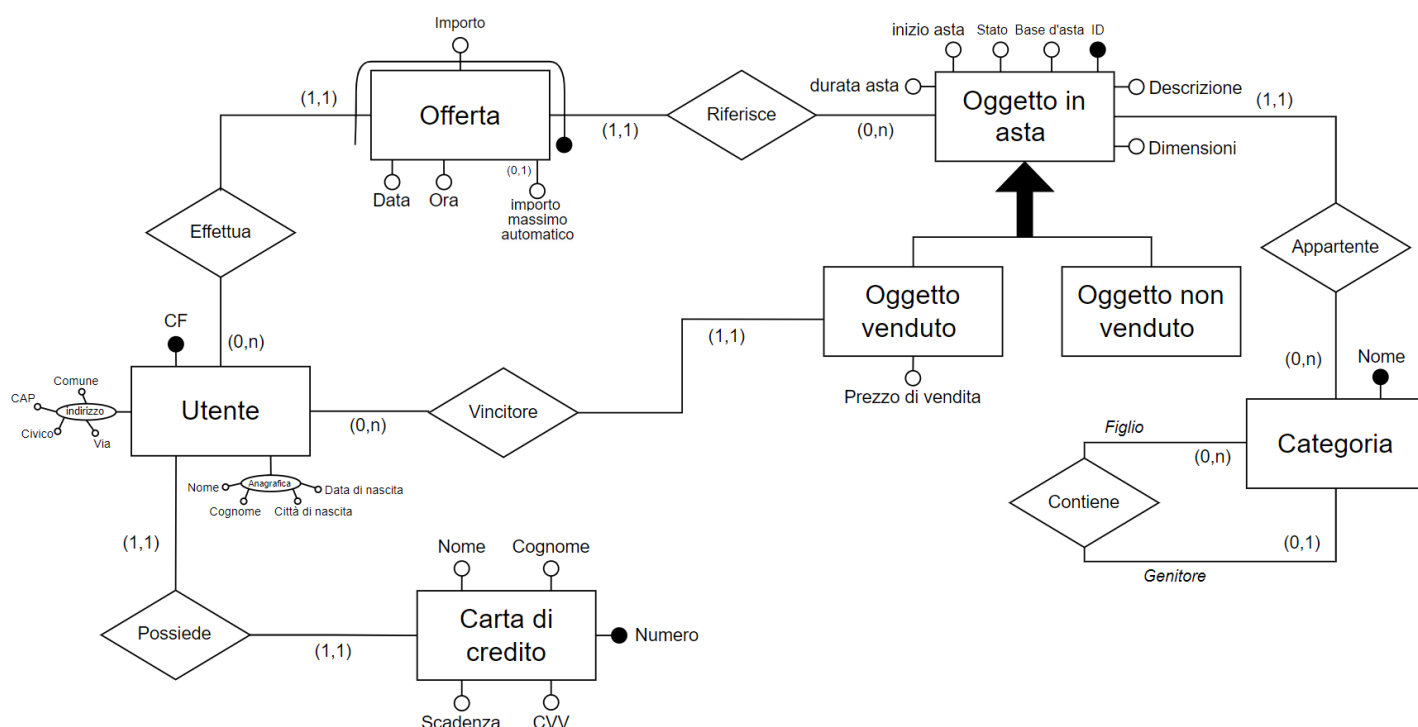


Quando un oggetto viene venduto, si vuole ovviamente tenere traccia di quale utente si è aggiudicato l'oggetto, per questo viene aggiunta la relazione "vincitore" tra Utente e Oggetto venduto. Utente partecipa alla relazione con cardinalità (0,n) mentre Oggetto venduto partecipa con cardinalità (1,1) poiché un oggetto è aggiudicato da uno ed un solo utente mentre un utente si può aggiudicare più oggetti. Inoltre, viene aggiunta l'entità Categoria, identificata dal Nome, associata ad Oggetto in asta tramite la relazione "appartiene". Per catturare il concetto che le categorie appartengono ad un titolare gerarchico, viene utilizzata una relazione ricorsiva "contiene", ogni categoria genitore ha più categorie figlie e ogni categoria figlia ha un solo padre, come in Figura 5.

FIGURA 5



## Integrazione finale



## Regole aziendali

- 1) Un Utente con carta di credito scaduta non deve poter effettuare un'offerta
- 2) Un Utente che si aggiudica un oggetto deve aver fatto l'offerta con il valore maggiore sull'oggetto in questione
- 3) Un Utente non deve poter effettuare un'offerta su un oggetto venduto
- 4) Un Utente non deve poter effettuare un'offerta su un oggetto per il quale è terminato il tempo d'asta.
- 5) Le Categorie devono avere al massimo 3 livelli gerarchici
- 6) Un'Offerta su un Oggetto deve avere un importo di 0,50 € superiore all'importo maggiore tra quello di tutte le offerte precedenti su quell'oggetto.
- 7) La prima Offerta su un Oggetto deve avere un importo di 0,50 € superiore a quello della base d'asta
- 8) La durata dell'asta deve essere compresa tra 1 e 7 giorni
- 9) La data di termine asta si ottiene da (data inizio asta + durata asta)
- 10) Il tempo mancante per il termine dell'asta si ottiene da ( data di termine asta - data corrente)
- 11) Il prezzo di vendita si ottiene dall'ultima offerta effettuata sull'oggetto in questione
- 12) L'importo dell'offerta massima automatica deve essere superiore al valore della base d'asta e deve rispettare la granularità di 0,50 €.

## Dizionario dei dati

Entità	Descrizione	Attributi	Identificatori
Oggetto in asta	Oggetto che viene messo in asta nel sistema	ID, Descrizione, Descrizione dimensioni, durata asta, Stato, inizio asta, Prezzo, Base d'asta	ID
Oggetto venduto	Oggetto per il quale è stata effettuata almeno un'offerta quando il tempo scade	Prezzo di vendita	ID
Oggetto non venduto	Oggetto per il quale non è stata effettuata alcuna offerta quando il tempo scade		ID
Offerta	Offerta con un determinato importo effettuata su un oggetto da un utente	Oggetto in asta, importo, Utente, Data, Ora, Importo massimo automatico	Oggetto in asta, importo, Utente
Utente	Colui che si registra nel sistema ed effettua offerte per gli oggetti in asta	CF, Anagrafica, Indirizzo	CF
Carta di Credito	Metodo di pagamento con il quale un utente paga l'oggetto aggiudicato	Numero, Nome, Cognome, CVV, Data di scadenza	Numero
Categoria	La categoria a cui appartiene un oggetto che viene messo in asta	Nome	Nome

Relazione	Descrizione	Entità coinvolte	Attributi
Riferisce	A quale Oggetto riferisce un'Offerta	Offerta, Oggetto in asta	
Effettua	Quale Utente effettua un'Offerta	Utente, Offerta	
Vincitore	Utente che si aggiudica un oggetto	Utente, Oggetto venduto	
Possiede	Utente che possiede una carta di credito	Utente, Carta di credito	
Appartiene	Oggetto in asta che appartiene ad una determinata categoria	Oggetto in asta, Categoria	
Contiene	Una categoria contiene una sottocategoria	Categoria Genitore, Categoria Figlia	

## 4. Progettazione logica

### Volume dei dati

Nell'analisi dei volumi si considera che i dati relativi alle aste vengano mantenuti nel sistema per un periodo di 1 anno dal termine.

- Si suppone che il numero di utenti sia circa 2000.
- Un utente compie in media 5 offerte per un oggetto in asta.
- Un utente effettua almeno un'offerta su circa il 5% degli oggetti in asta.
- Gli oggetti in asta sono circa 1500.
- Circa il 70% degli oggetti in asta viene venduto, il 30% non viene venduto.
- Il numero di categorie è 90.
- Calcolo offerte:
  - Si suppone che ogni utente faccia offerte sul 5% degli oggetti in asta. Quindi, ogni utente farebbe offerte su  $1500 \text{ oggetti} * 0.05 = 75 \text{ oggetti}$
  - Supponiamo inoltre che per ogni oggetto, ogni utente faccia in media 5 offerte.
  - Offerte =  $2000 \text{ utenti} * 75 \text{ oggetti/utente} * 5 \text{ offerte/oggetto} = 750.000 \text{ offerte}$

Concetto nello schema	Tipo <sup>1</sup>	Volume atteso
Oggetto in asta	E	1500
Oggetto venduto	E	1100
Oggetto non venduto	E	400
Offerta	E	750.000
Utente	E	2000
Carta di Credito	E	2000
Categoria	E	90
Riferisce	R	750.000
Effettua	R	750.000
Vincitore	R	1100
Possiede	R	2000
Appartiene	R	1500
Contiene	R	90

---

<sup>1</sup> Indicare con E le entità, con R le relazioni

## Tavola delle operazioni

Cod.	Descrizione	Frequenza attesa
U1	L'utente effettua un'offerta per un oggetto	200 / giorno
U2	L'utente imposta la controfferta automatica	50 / settimana
U3	L'utente visualizza le aste non concluse a cui ha fatto un'offerta	20 / giorno
U4	L'utente visualizza gli oggetti delle aste vinte	10 / giorno
U5	L'utente visualizza le aste aperte	40 / giorno
L1	Login	50 / giorno
R1	Registrazione	8 / settimana
A1	L'Amministratore inserisce un Oggetto in asta	4 / giorno
A2	L'Amministratore inserisce una Categoria	3 / anno
A3	L'Amministratore modifica una Categoria	4 / anno
A4	L'Amministratore elimina una Categoria	2 / anno

## Costo delle operazioni

Si suppone, per il calcolo del costo delle operazioni, che il costo delle scritture sia il doppio del costo delle letture.

*Operazione U1 : L'utente effettua un'offerta per un oggetto*

- Accessi in lettura :
  - Utente [E] : 1 accesso in lettura
  - Oggetto in asta [E] : 1 accesso in lettura
  - Possiede [R] : 1 accesso in lettura
  - Carta di credito [E] : 1 accesso in lettura
  - Riferisce [R] : 500 accessi in lettura
  - Offerta [E] : 500 accessi in lettura
- Accessi in scrittura :
  - Effettua [R] : 1 accesso in scrittura
  - Riferisce [R] : 1 accesso in scrittura
  - Offerta [E] : 1 accesso in scrittura

Totale costo dell'operazione :  $1 + 1 + 1 + 1 + 500 + 500 + 2 + 2 + 2 = 1010$

Totale costo (costo operazione \* frequenza operazione) =  $10 + 2 * 500 * 200 = 202.000$  accessi / g

*Operazione U2 : L'utente imposta la controfferta automatica*

- Accessi in lettura :
  - Utente [E] : 1 accesso in lettura
  - Oggetto in asta [E] : 1 accesso in lettura
  - Possiede [R] : 1 accesso in lettura
  - Carta di credito [E] : 1 accesso in lettura
  - Riferisce [R] : 500 accessi in lettura
  - Offerta [E] : 500 accessi in lettura
- Accessi in scrittura :
  - Effettua [R] : 1 accesso in scrittura
  - Riferisce [R] : 1 accesso in scrittura
  - Offerta [E] : 1 accesso in scrittura

Totale costo dell'operazione :  $1 + 1 + 1 + 1 + 500 + 500 + 2 + 2 + 2 = 1010$

Totale costo (costo operazione \* frequenza operazione ) =  $(10 + 2 * 500) * 50 = 50.500$  accessi / sett

*Operazione U3 : L'utente visualizza le aste non concluse a cui ha fatto un'offerta*

- Accessi in lettura :
  - Utente [E] : 1 accesso in lettura
  - Oggetto in asta [E] : 1 accesso in lettura
  - Effettua [R] : 1000 accessi in lettura
  - Riferisce [R] : 1000 accessi in lettura
  - Offerta [E] : 1000 accessi in lettura

Totale costo dell'operazione :  $1 + 1 + 1000 + 1000 + 1000 = 3.002$

Totale costo (costo operazione \* frequenza operazione ) =  $(2 + 3.000) * 20 = 60.040$  accessi / g

*Operazione U4 : L'utente visualizza gli oggetti delle aste che ha vinto*

- Accessi in lettura :
  - Utente [E] : 1 accesso in lettura
  - Oggetto Venduto [E] : 1 accesso in lettura
  - Vincitore [R] : 1 accesso in lettura

Totale costo dell'operazione :  $1 + 1 + 1 = 3$

Totale costo (costo operazione \* frequenza operazione ) =  $3 * 10 = 30$  accessi / g

*Operazione U5 : L'utente visualizza le aste aperte*

- Accessi in lettura :
  - Oggetto in asta [E]            20 accessi in lettura
  - Offerta [E] :            16.000 accessi in lettura
  - Riferimento [R] :    16.000 accessi in lettura

Totale costo dell'operazione :  $40 + 16.000 + 16.000 = 32.020$

Totale costo (costo operazione \* frequenza operazione ) =  $32.020 * 40 = 1.280.800$  accessi / g

*Operazione L1 : Login*

- Accessi in lettura :
  - Utente [E]                    1 accesso in lettura

Totale costo (costo operazione \* frequenza operazione ) =  $1 * 50 = 50$  accessi / g

*Operazione R1 : Registrazione*

- Accessi in scrittura :
  - Utente [E]                    1 accesso in scrittura

Totale costo (costo operazione \* frequenza operazione ) =  $2 * 8 = 16$  accessi / sett

*Operazione A1 : L'Amministratore inserisce un Oggetto in asta*

- Accessi in scrittura :
  - Oggetto in asta [E]            1 accesso in scrittura

Totale costo (costo operazione \* frequenza operazione ) =  $2 * 4 = 8$  accessi / g

*Operazione A2 : L'Amministratore inserisce una Categoria*

- Accessi in lettura :
  - Contiene [R]                    3 accessi in lettura
- Accessi in scrittura :
  - Categoria [E]                    3 accessi in scrittura
  - Contiene [R]                    1 accesso in scrittura

Totale costo dell'operazione :  $3 + 6 + 2 = 11$

Totale costo (costo operazione \* frequenza operazione ) =  $11 * 3 =$  accessi / anno



*Operazione A3 : L'Amministratore modifica una Categoria*

- Accessi in lettura :
  - Contiene [R]                      1 accesso in lettura
  - Categoria [E]                    1 accesso in lettura
- Accessi in scrittura :
  - Categoria [E]                    1 accesso in scrittura
  - Contiene [R]                    1 accesso in scrittura

Totale costo dell'operazione :  $1 + 1 + 2 + 2 = 6$

Totale costo (costo operazione \* frequenza operazione ) =  $6 * 4 = 24$  accessi / anno

*Operazione A4 : L'Amministratore elimina una Categoria*

- Accessi in lettura :
  - Contiene [R]                      1 accesso in lettura
  - Categoria [E]                    1 accesso in lettura
- Accessi in scrittura :
  - Categoria [E]                    1 accesso in scrittura
  - Contiene [R]                    1 accesso in scrittura

Totale costo dell'operazione :  $1 + 1 + 2 + 2 = 6$

Totale costo (costo operazione \* frequenza operazione ) =  $6 * 2 = 12$  accessi / anno

**Ristrutturazione dello schema E-R***1) Analisi delle ridondanze*

In questa sezione si vuole prendere in esame i dati che possono essere ricavati tramite operazioni tra attributi come “Numero Offerte”, “Valore offerta Massima” e “Termine asta”, facendo delle considerazioni su prestazioni e gestione della memoria con il fine di decidere se mantenere, aggiungere o rimuovere delle ridondanze.

Si è scelto di introdurre la ridondanza riguardante “Numero Offerte” e “Valore offerta Massima”. Questa decisione è stata presa per migliorare le prestazioni del sistema, evitando di dover contare ogni volta che viene fatta un'offerta tutte le occorrenze delle offerte e tutti gli importi di esse riferite ad uno specifico oggetto. Questa scelta comporta tuttavia un maggior utilizzo della memoria. Consideriamo che il numero di oggetti in asta sia 1500, un tipo di dato intero per il

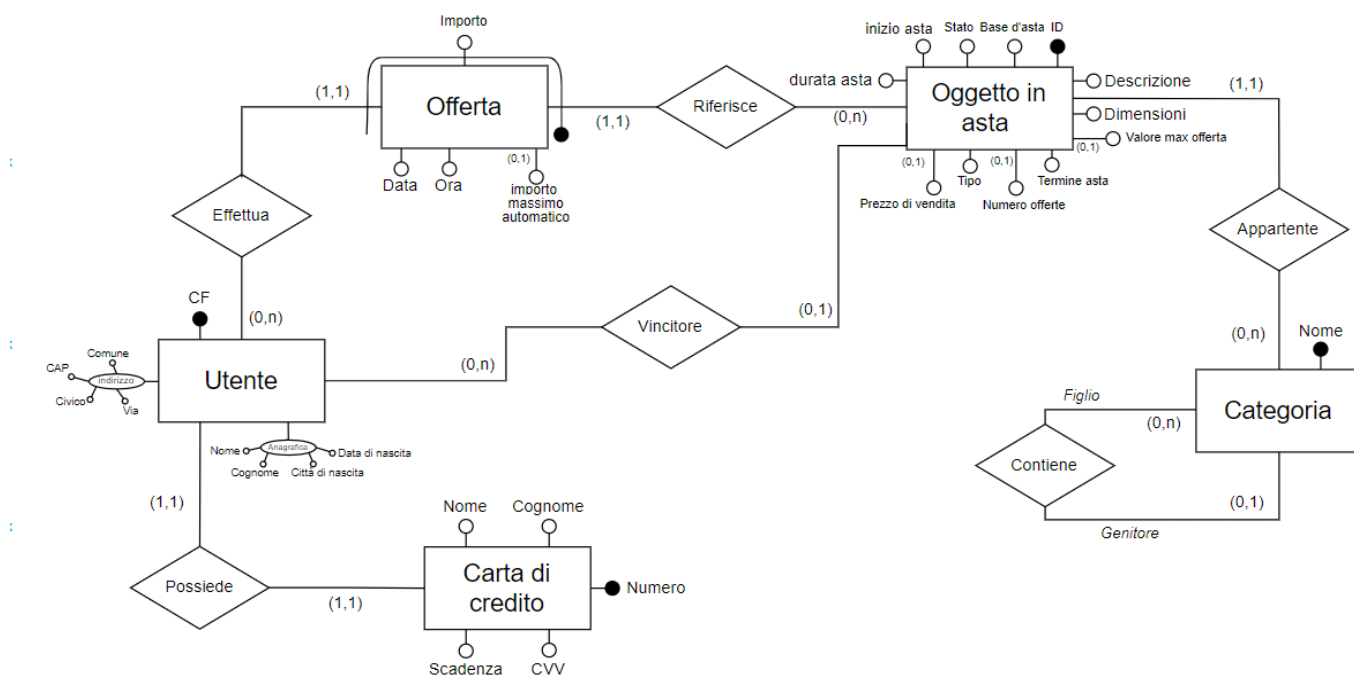
numero di offerte (4 B) e il valore dell'offerta più alta un dato di tipo virgola mobile (8 B), in totale avremmo  $1500 * 4 + 1500 * 8 = 6.000 + 12.000 = 18 \text{ kB}$  in più per l'entità Oggetto in asta. Vediamo un esempio considerando i costi delle operazioni U1, U2 e U5 che sono molto simili nel concetto. Per esempio, l'operazione U1, che rappresenta un utente che effettua un'offerta per un oggetto, richiede 500 accessi in lettura per "Riferisce" e "Offerta". Introducendo la ridondanza, si può ridurre significativamente il numero di accessi in lettura, avremmo tuttavia un accesso in scrittura in più per "Effettua", "Riferisce" e "Offerta". In conclusione, la decisione di introdurre la ridondanza è una scelta strategica per migliorare le prestazioni del sistema a discapito di un maggiore utilizzo della memoria.

Per quanto riguarda " Termine asta " possiamo fare lo stesso ragionamento fatto in precedenza. Il momento in cui termina l'asta può essere ottenuto sommando all'inizio dell'asta il valore che riferisce alla durata di essa. Le operazioni in cui influiscono tali attributi sono U5 e A1 che riguardano inserimento e visualizzazione degli oggetti in asta. Se la data di fine asta fosse mantenuta in memoria con un tipo di dato stringa di 10 caratteri GG/MM/AAAA avremmo  $10 * 1.500 = 15 \text{ kB}$  di memoria utilizzata in più per l'entità Oggetto in asta. Si decide quindi di introdurre la ridondanza per non appesantire l'operazione di visualizzazione degli oggetti a discapito di un utilizzo maggiore di memoria.

## 2) *Eliminazione delle generalizzazioni*

Lo schema ER precedentemente mostrato presenta una sola generalizzazione, ovvero Oggetto in asta con figli Oggetto venduto ed Oggetto non venduto. Si è deciso di accorpare le entità figlie nell'entità genitore. Le operazioni non fanno molta distinzione tra le occorrenze e tra gli attributi delle entità coinvolte. In questo caso infatti, anche se abbiamo uno spreco di memoria per la presenza di valori nulli, la scelta ci assicura un numero minore di accessi rispetto alle altre tecniche di eliminazione nelle quali le occorrenze e gli attributi sono distribuiti tra le varie entità. Nello specifico viene aggiunto all'entità Oggetto in asta l'attributo opzionale " prezzo di vendita" e l'attributo "Tipo" che identifica lo stato di vendita dell'oggetto. Inoltre siccome l'entità Oggetto venduto era associata all'entità Utente tramite la relazione "Vincitore", essa viene associata direttamente all'entità Oggetto in asta cambiando però la partecipazione alla relazione da (1,1) a (0,1) diventando quindi opzionale

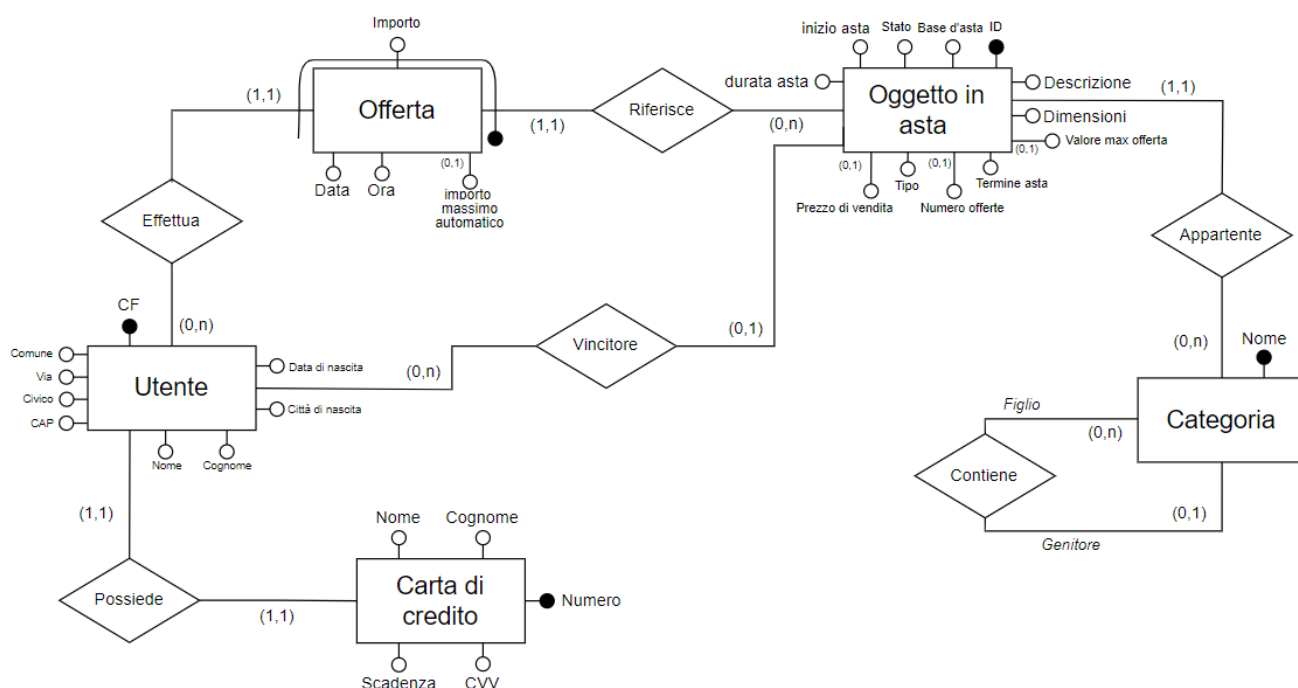
Di seguito lo schema Entità-Relazione ristrutturato



## Trasformazione di attributi e identificatori

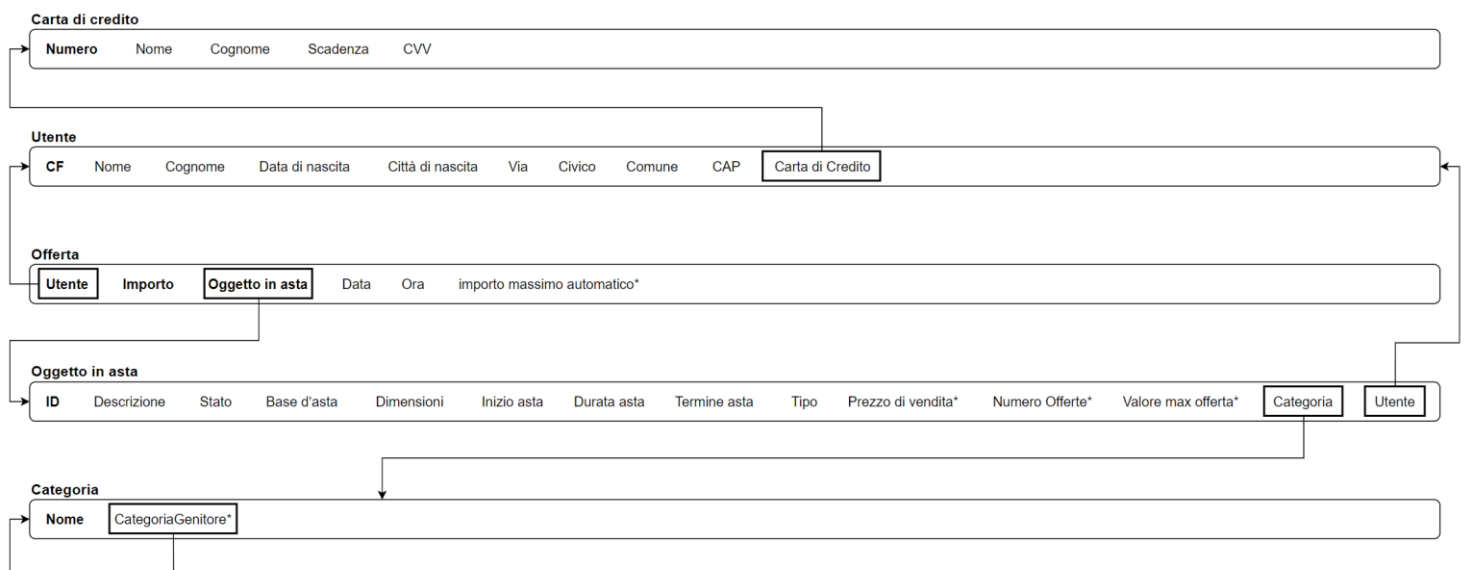
In questa sezione vengono rimossi gli attributi composti al fine di rendere più intuitiva la trasformazione in modello relazionale. Vengono quindi rimossi gli attributi composti “indirizzo” ed “anagrafica” dell’entità Utente.

Di seguito lo schema Entità-Relazione ristrutturato in seguito alla trasformazione degli attributi



## Traduzione di entità e associazioni

- Utente ( CF , Nome , Cognome , Data di nascita , Città di nascita , Via , Civico , Comune , CAP, Carta di Credito )
  - Carta di Credito ( Numero , Nome, Cognome, Scadenza, CVV)
  - Offerta ( Utente, Importo, Oggetto in asta, Data, Ora, importo massimo automatico\*)
  - Oggetto in asta ( ID , Descrizione , Stato , Base d'asta , Dimensioni , Inizio asta , Durata asta , Termine asta, Tipo , Prezzo di vendita\* , Numero Offerte\* , Valore max offerta\* , Categoria , Utente\*)
  - Categoria ( Nome , CategoriaGenitore\* )
- 
- $\text{Utente}(\text{Carta di Credito}) \subseteq \text{Carta di Credito}(\text{Numero})$
  - $\text{Offerta}(\text{Utente}) \subseteq \text{Utente}(\text{CF})$
  - $\text{Offerta}(\text{Oggetto in asta}) \subseteq \text{Oggetto in asta}(\text{ID})$
  - $\text{Oggetto in asta}(\text{Categoria}) \subseteq \text{Categoria}(\text{Nome})$
  - $\text{Oggetto in asta}(\text{Utente}) \subseteq \text{Utente}(\text{CF})$
  - $\text{Categoria}(\text{CategoriaGenitore*}) \subseteq \text{Categoria}(\text{Nome})$



## Normalizzazione del modello relazionale

- 1NF

Una relazione soddisfa la 1NF se gli attributi delle relazioni sono definiti su valori atomici e non su valori complessi quali insiemi o relazioni. Osserviamo che nel caso in questione lo schema è in 1NF perché ogni colonna delle tabelle contiene valori indivisibili e uniformi, senza ripetizioni o gruppi di valori.

- 2NF

Una relazione è in 2NF se è in 1NF e se su di essa non sono definite dipendenze parziali, cioè dipendenze fra un sottoinsieme proprio della chiave e altri attributi. Lo schema in questione è in 2NF perché ogni tabella ha una chiave primaria ben definita e tutti gli attributi non chiave sono completamente dipendenti dalla chiave primaria,

- 3NF

Una relazione è in 3NF se è in 2NF e se su di essa non sono definite dipendenze transitive. Lo schema in questione è in 3NF perché non ci sono dipendenze transitive tra gli attributi non chiave. Ogni attributo non chiave dipende solo dalla chiave primaria.

## 5. Progettazione fisica

### Utenti e privilegi

Si prevedono tre ruoli, per implementare il Principle of Least Privilege:

- Login:
  - Grant in esecuzione sull'operazione L1 e R1
- Utente
  - Grant in esecuzione sulle operazioni U1, U2, U3, U4, U5
- Amministratore
  - Grant in esecuzione sulle operazioni A1, A2, A3, A4

Per identificare gli amministratori si introduce una tabella Amministratori per mantenerne le credenziali.

### Strutture di memorizzazione

Tabella Offerta		
Colonna	Tipo di dato	Attributi <sup>2</sup>
Utente	VARCHAR(16)	PK, NN
Importo	FLOAT	PK, NN
Oggetto in asta	VARCHAR(10)	PK, NN
Data	DATE	NN
Ora	TIME	NN
Importo massimo automatico	FLOAT	

Tabella Utente		
Colonna	Tipo di dato	Attributi <sup>2</sup>
CF	VARCHAR(16)	PK, NN
Nome	VARCHAR(45)	NN
Cognome	VARCHAR(45)	NN
Data di nascita	DATE	NN
Città di nascita	VARCHAR(45)	NN
Via	VARCHAR(45)	NN
Civico	VARCHAR(6)	NN
Comune	VARCHAR(45)	NN
Carta di credito	VARCHAR(16)	NN
Username	VARCHAR(45)	NN, UQ
Password	VARCHAR(45)	NN

---

<sup>2</sup> PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna.

Tabella Carta di credito		
Colonna	Tipo di dato	Attributi <sup>2</sup>
Numero	VARCHAR(16)	PK, NN
Nome	VARCHAR(45)	NN
Cognome	VARCHAR(45)	NN
Scadenza	VARCHAR(5)	NN
CVV	VARCHAR(4)	NN

Tabella Oggetto in asta		
Colonna	Tipo di dato	Attributi <sup>2</sup>
ID	VARCHAR(10)	PK, NN
Descrizione	VARCHAR(45)	NN
Stato	VARCHAR(45)	NN
Base d'asta	FLOAT	NN
Dimensioni	VARCHAR(45)	NN
Inizio asta	DATETIME	NN
Durata asta	INT	NN
Termine asta	DATE	NN
Tipo	ENUM('in corso', 'venduto', 'non venduto')	NN
Prezzo di vendita	FLOAT	
Numero offerte	INT	
Valore max offerta	FLOAT	
Categoria	VARCHAR(45)	NN
Utente	VARCHAR(16)	

Tabella Categoria		
Colonna	Tipo di dato	Attributi <sup>2</sup>
Nome	VARCHAR(45)	PK, NN
CategoriaGenitore	VARCHAR(45)	

Tabella Amministratore		
Colonna	Tipo di dato	Attributi <sup>2</sup>
Username	VARCHAR(45)	PK, NN
Password	VARCHAR(45)	NN

## Indici

Si omettono gli indici di tipo PRIMARY per le chiavi e INDEX per le foreign key, in quanto generati automaticamente.

Tabella Offerta	
idx_offerta_importo	Tipo
Importo	IDX

L'indice su Importo nella tabella Offerta migliora le performance delle operazioni U2 e U1, che effettuano frequenti confronti sugli importi delle offerte. Questo indice rende più veloci le operazioni di verifica degli importi, assicurando che le nuove offerte siano valide rispetto a quelle esistenti. Così facendo, si migliora l'efficienza e l'integrità delle transazioni d'asta.

Tabella Oggetto in asta	
idxoggetto_in_asta_termine_asta	Tipo
Termine asta	IDX

L'indice su Termine asta nella tabella Oggetto in asta migliora le performance delle operazioni U3 e U5. Questo indice rende più rapide le operazioni di ricerca delle aste attive e la visualizzazione delle aste che stanno per concludersi, migliorando così l'efficienza delle operazioni.

## Trigger

- Trigger per il controllo del livello gerarchico di una categoria prima dell'inserimento

```
CREATE DEFINER='root'@'localhost' TRIGGER `check_category_level` BEFORE INSERT ON `categoria` FOR EACH ROW BEGIN
    DECLARE level INT DEFAULT 0;
    DECLARE parent_category_name VARCHAR(45);

    SET parent_category_name = NEW.categoriaGenitore;

    -- Ciclo per verificare il livello della gerarchia
    WHILE parent_category_name IS NOT NULL DO
        SET level = level + 1;

        -- Recupera il genitore della categoria corrente
        SELECT macrocategoria INTO parent_category_name
        FROM categoria
        WHERE nome = parent_category_name;

        -- Verifica se il livello supera il massimo consentito
        IF level > 2 THEN
            SIGNAL SQLSTATE '45020' SET MESSAGE_TEXT = 'Error: The category exceeds the maximum allowed hierarchy level (3 levels).';
        END IF;
    END WHILE;
END
```



- Trigger per il controllo del livello gerarchico di una categoria prima dell'aggiornamento

```
CREATE DEFINER='root'@'localhost' TRIGGER `check_category_level_update` BEFORE UPDATE ON `categoria` FOR EACH ROW BEGIN
    DECLARE level INT DEFAULT 0;
    DECLARE parent_category_name VARCHAR(45);

    SET parent_category_name = NEW.categoriaGenitore;

    -- Ciclo per verificare il livello della gerarchia
    WHILE parent_category_name IS NOT NULL DO
        SET level = level + 1;

        -- Recupera il genitore della categoria corrente
        SELECT macrocategoria INTO parent_category_name
        FROM categoria
        WHERE nome = parent_category_name;

        -- Verifica se il livello supera il massimo consentito
        IF level > 2 THEN
            SIGNAL SQLSTATE '45020' SET MESSAGE_TEXT = 'Error: The category exceeds the maximum allowed hierarchy level (3 levels).';
        END IF;
    END WHILE;
END
```

- Trigger per controllare che la durata dell'asta sia compresa tra 1 e 7 giorni

```
CREATE DEFINER='root'@'localhost' TRIGGER `aste`.`check_auction_duration` BEFORE INSERT ON `aste`.`oggetto in asta`
FOR EACH ROW
BEGIN
    IF NEW.`durata asta` < 1 OR NEW.`durata asta` > 7 THEN
        SIGNAL SQLSTATE '45030'
        SET MESSAGE_TEXT = 'The duration of the auction must be between 1 and 7 days';
    END IF;
    IF DATEDIFF(NEW.`termine asta`,NEW.`inizio asta`) != NEW.`durata asta` THEN
        SIGNAL SQLSTATE '45031'
        SET MESSAGE_TEXT = ' The duration of the auction does not match with start and finish date';
    END IF;
END
```

## Eventi

- Evento per la cancellazione dei dati riguardanti le aste che siano terminate da più di 1 anno.

```
CREATE EVENT IF NOT EXISTS `delete_old_auctions`
ON SCHEDULE EVERY 1 YEAR STARTS CURRENT_TIME + INTERVAL 1 YEAR ON COMPLETION PRESERVE
COMMENT "Delete ended auctions older than 1 year"
DO
BEGIN
    -- Cancella oggetti in asta terminati più di 1 anno fa
    DELETE FROM `oggetto in asta`
    WHERE `Termine asta` < DATE_SUB(NOW(), INTERVAL 1 YEAR);
END;
```

- Evento che cambia il tipo di asta, ovvero lo stato di proseguimento dell'asta, nel momento in cui il tempo di durata dell'asta è scaduto

```

CREATE EVENT IF NOT EXISTS `update_auction_status`
ON SCHEDULE EVERY 1 MINUTE
STARTS CURRENT_TIME ON COMPLETION PRESERVE
DO
BEGIN

    DECLARE var_user          VARCHAR(16);
    DECLARE var_id            VARCHAR(10);
    DECLARE var_max_offerta   FLOAT;
    DECLARE var_numero_offerte INT;

    -- Oggetto per il quale il giorno di fine asta e' successivo al giorno corrente
    SELECT ID, `Numero offerte`, `Valore max offerta`
    FROM   `oggetto in asta`
    WHERE  current_date() > `Termine asta`
    AND    Tipo = 'in corso'

    UNION

    -- Oggetto per il quale il giorno di fine asta e' il giorno corrente, ma l'orario di fine e' superato
    SELECT ID, `Numero offerte`, `Valore max offerta`
    FROM   `oggetto in asta`
    WHERE  current_date() = `Termine asta`
    AND    current_time() > time(`Inizio asta`)
    AND    Tipo = 'in corso'

    LIMIT 1

    INTO var_id, var_numero_offerte, var_max_offerta;

    IF var_numero_offerte is null then
    -- Se non ci sono state offerte aggiorna ad oggetto non venduto
        UPDATE `oggetto in asta`
        SET     Tipo = 'non venduto'
        WHERE  ID = var_id;

    ELSEIF var_numero_offerte >= 1 then
    -- Se ci sono state offerte aggiorna ad oggetto venduto ed imposta importo ed acquirente
        SELECT offerta.Utente
        FROM   offerta
        JOIN   `oggetto in asta` ON ID = offerta.`oggetto in asta`
        WHERE  offerta.Importo = var_max_offerta
        INTO   var_user;

        UPDATE `oggetto in asta`
        SET     Tipo = 'venduto', `Prezzo di Vendita` = var_max_offerta, Utente = var_user
        WHERE  ID = var_id;

    END IF;

END

```

## Viste

Non sono state utilizzate viste

## Funzioni

- Funzione utilizzata per controllare la corretta forma del Codice Fiscale

```
CREATE DEFINER='root'@'localhost' FUNCTION `check_cf`(CF varchar(16)) RETURNS tinyint(1)
    DETERMINISTIC
BEGIN
    IF CF regexp '([A-Z]{6}[0-9LMNPQRSTU]{2}[ABCDEHLMPRST]{1}[0-9LMNPQRSTU]{2}[A-Z]{1}[0-9LMNPQRSTU]{3}[A-Z]{1})$|([0-9]{11})$' then
        return true;
    ELSE
        return false;
    END IF;
END
```

- Funzione utilizzata per controllare la corretta forma del CAP

```
CREATE DEFINER='root'@'localhost' FUNCTION `check_cap`( CAP varchar(5) ) RETURNS tinyint(1)
    DETERMINISTIC
BEGIN
    IF CAP regexp '^([0-9]{5})$' then
        return true ;
    ELSE
        return false ;
    END IF;
END
```

- Funzione utilizzata per controllare la corretta forma della Carta di Credito

```
CREATE DEFINER='root'@'localhost' FUNCTION `check_credit_card`(creditcard varchar(16)) RETURNS tinyint(1)
    DETERMINISTIC
BEGIN
    if creditcard regexp '([0-9]{16})$' then
        return true;
    else
        return false;
    END IF ;
END
```

- Funzione utilizzata per controllare la corretta forma della data di Scadenza della Carta di Credito

```
CREATE DEFINER='root'@'localhost' FUNCTION `check_expire_date`(scadenza varchar(5)) RETURNS tinyint(1)
    DETERMINISTIC
BEGIN
    if scadenza regexp '([01-9]|1[0-2])-[0-9]{2}$' then
        return true;
    else
        return false;
    end if;
END
```

- Funzione utilizzata per controllare la corretta forma del CVV, consideriamo sia CVC (4 cifre) che il CVV (3 cifre)

```
CREATE DEFINER='root'@'localhost' FUNCTION `check_cvv`( cvv varchar(4) ) RETURNS tinyint(1)
    DETERMINISTIC
BEGIN
    IF cvv regexp '^[0-9]{3,4}$' then
        return true;
    ELSE
        return false;
    END IF;
END
```

- Funzione utilizzata per controllare che la Carta di Credito non sia Scaduta

```
CREATE DEFINER='root'@'localhost' FUNCTION `check_valid_expire`(scadenza varchar(5)) RETURNS tinyint(1)
    DETERMINISTIC
BEGIN
    DECLARE expiry_date DATE;
    DECLARE curr_date DATE;

    -- Converte la data di scadenza nel primo giorno del mese successivo
    SET expiry_date = STR_TO_DATE(CONCAT('01-', scadenza), '%d-%m-%y');

    -- Ottiene la data corrente
    SET curr_date = CURDATE();

    -- Confronta le date
    IF expiry_date < curr_date THEN
        RETURN false; -- Carta scaduta
    ELSE
        RETURN true; -- Carta valida
    END IF;
END
```

- Funzione utilizzata per controllare la corretta forma del Numero Civico di spedizione

```
CREATE DEFINER='root'@'localhost' FUNCTION `check_civico`(civico varchar(6)) RETURNS tinyint(1)
    DETERMINISTIC
BEGIN
    if civico regexp '^[0-9]{1,5}[a-zA-Z]{0,1}$' then
        return true;
    else
        return false;
    end if ;
END
```

## Stored Procedures e transazioni

- **login**: Procedura che implementa l'operazione di login

```
CREATE DEFINER='root'@'localhost' PROCEDURE `login`(in var_username VARCHAR(45), in var_password VARCHAR(45), out var_role INT)
BEGIN

    DECLARE is_admin INT;
    DECLARE is_user INT;

    SELECT COUNT(*) INTO is_admin
    FROM aste.amministratore
    WHERE username = var_username AND password = md5(var_password);

    SELECT COUNT(*) INTO is_user
    FROM aste.utente
    WHERE username = var_username AND password = md5(var_password);

    IF is_user = 1 THEN
        SET var_role = 1;
    ELSEIF is_admin = 1 THEN
        SET var_role = 2;
    ELSE
        SET var_role = 3;
    END IF;
END
```

- **admin\_registration\_procedure**: Questa procedura viene utilizzata per registrare un nuovo amministratore. Si è scelto il livello di isolamento Serializable per evitare che due transazioni simultanee creino duplicati del nome utente, garantendo così l'integrità dei dati nel database.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `admin_registration_procedure`( in var_username VARCHAR(45), in var_password VARCHAR(45) )
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
    START TRANSACTION;

    -- Controllo se il username esiste già nella tabella utente
    IF (SELECT COUNT(*) FROM utente WHERE username = var_username) > 0 THEN
        SIGNAL SQLSTATE '45040' SET MESSAGE_TEXT = "Username already exists ";
    END IF;
    INSERT INTO amministratore(username, `password`) VALUES (var_username, md5(var_password));

    COMMIT;
END
```

- **get\_purchased\_items**: Procedura che mostra ad un utente gli oggetti per i quali ha vinto l'asta.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `get_purchased_items`(in var_user VARCHAR(16))
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
    START TRANSACTION;

    SELECT ID, Descrizione, Stato, Dimensioni, `Termine asta`, `Prezzo di vendita`, Categoria
    FROM `oggetto in asta`
    WHERE Utente = var_user
    AND Tipo = 'venduto';

    COMMIT;
END
```

- **user\_registration\_procedure** : Questa procedura viene utilizzata per la registrazione di un nuovo utente, inizialmente si effettuano i vari controlli per garantire che i dati vengano inseriti correttamente, in seguito viene inserito il nuovo utente. Come per la registrazione dell'amministratore si utilizza il livello di isolamento Serializable per evitare username duplicati.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `user_registration_procedure`( in var_cf VARCHAR(16), in var_name VARCHAR(45), in var_surname VARCHAR(45), in var_birth_date DATE,
in var_birth_city VARCHAR(45), in var_address VARCHAR(45), in var_civico VARCHAR( 6), in var_comune VARCHAR(45), in var_cap VARCHAR( 5), in var_credit_card VARCHAR(16),
in var_holder_name VARCHAR(45), in var_holder_surname VARCHAR(45), in var_expire_date VARCHAR(5), in var_cvv VARCHAR(4), in var_username VARCHAR(45), in var_password VARCHAR(45))
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;
    -- Si effettuano i vari controlli per il corretto inserimento dei dati
    SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
    START TRANSACTION;

    -- Controllo se il username esiste già nella tabella amministratore
    IF (SELECT COUNT(*) FROM amministratore WHERE username = var_username) > 0 THEN
        SIGNAL SQLSTATE '45040' SET MESSAGE_TEXT = "Username already exists";
    END IF;

    IF(check_cf(var_cf) is false) then
        SIGNAL SQLSTATE '45001' SET message_text = " Invalid Codice Fiscale ";
    END IF;

    IF(check_civico(var_civico) is false) then
        SIGNAL SQLSTATE '45002' SET message_text = " Invalid House Number ";
    END IF;

    IF(check_cap(var_cap) is false) then
        SIGNAL SQLSTATE '45003' SET message_text = " Invalid Postal Code ";
    END IF;

    IF(check_credit_card(var_credit_card) is false) then
        SIGNAL SQLSTATE '45004' SET message_text = " Invalid Credit Card Number ";
    END IF;

    IF(check_cvv(var_cvv) is false) then
        SIGNAL SQLSTATE '45005' SET message_text = " Invalid CVV ";
    END IF;

    IF(check_expire_date(var_expire_date) is false) then
        SIGNAL SQLSTATE '45006' SET message_text = " Invalid Expire Date ";
    END IF;

    IF(check_valid_expire(var_expire_date) is false) then
        SIGNAL SQLSTATE '45007' SET message_text = " Expired Credit Card ";
    END IF;

    INSERT INTO `Carta di Credito`( Numero, Nome, Cognome, Scadenza, CVV )
    VALUES ( var_credit_card, var_holder_name, var_holder_surname, var_expire_date, var_cvv );

    INSERT INTO Utente(CF, Nome, Cognome, `Data di Nascita`, `Città di nascita`, Via, Civico, Comune, CAP, `Carta di Credito`, Username, `Password`)
    VALUES(var_cf, var_name, var_surname, var_birth_date, var_birth_city, var_address, var_civico, var_comune, var_cap, var_credit_card, var_username, md5(var_password) );

    COMMIT;
END
```

- **get\_user\_procedure**: Procedura per ottenere i dati utili di un utente una volta loggato

```
CREATE DEFINER='root'@'localhost' PROCEDURE `get_user_procedure`(in var_username VARCHAR(45))
BEGIN
    SELECT cf, nome, cognome, `data di nascita`,
    `città di nascita`,via,civico,comune,cap, `carta di credito`
    FROM utente
    WHERE username = var_username;
END
```



- **insert\_item\_procedure:** Procedura per l'inserimento di un nuovo oggetto in asta da parte di un amministratore. Siccome non si evidenziano anomalie rilevanti si utilizza Read Uncommitted.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `insert_item_procedure`(
    in var_description VARCHAR(45),
    in var_stato VARCHAR(45),
    in var_base FLOAT,
    in var_dim VARCHAR(45),
    in var_duration INTEGER,
    in var_category VARCHAR(45))
BEGIN
    DECLARE var_id VARCHAR(10);
    DECLARE var_start DATETIME;
    DECLARE var_finish DATE;
    DECLARE var_type ENUM('in corso', 'venduto', 'non venduto');
    DECLARE var_offer_n INTEGER;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
    START TRANSACTION ;

    SET var_id = SUBSTRING(REPLACE(UUID(), '-', ''), 1, 10);
    SET var_start = current_timestamp();
    SET var_finish = DATE_ADD(var_start, interval var_duration day);
    SET var_type = 'in corso';
    SET var_offer_n = 0;

    INSERT INTO `oggetto in asta` (ID, Descrizione, Stato, `Base d'asta`, Dimensioni, `Inizio asta`, `Durata asta`, `Termine asta`, Tipo, `Numero offerte`, Categoria)
    VALUES (var_id, var_description, var_stato, var_base, var_dim, var_start, var_duration, var_finish, var_type, var_offer_n, var_category);

    COMMIT;
END
```

- **get\_offered\_items:** Procedura per mostrare ad un utente gli oggetti ancora in asta per i quali ha effettuato un'offerta.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `get_offered_items`( in var_user VARCHAR(16) )
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
    START TRANSACTION;

    SELECT ID, Descrizione, Stato, `Base d'asta`, Dimensioni, `Inizio asta`, `Termine asta`, `Numero offerte`, `Valore max offerta`, Categoria
    FROM `oggetto in asta`
    JOIN offerta as O ON `oggetto in asta`.ID = O.`Oggetto in asta`
    WHERE `oggetto in asta`.tipo = 'in corso'
    AND `Termine asta` > CURRENT_DATE()
    AND var_user IN (SELECT Utente FROM offerta WHERE O.Utente = offerta.Utente )
    UNION
    SELECT ID, Descrizione, Stato, `Base d'asta`, Dimensioni, `Inizio asta`, `Termine asta`, `Numero offerte`, `Valore max offerta`, Categoria
    FROM `oggetto in asta`
    JOIN offerta as O ON `oggetto in asta`.ID = O.`Oggetto in asta`
    WHERE `oggetto in asta`.tipo = 'in corso'
    AND `Termine asta` = CURRENT_DATE()
    AND `Inizio asta` - CURRENT_TIME > 0
    AND var_user IN (SELECT Utente FROM offerta WHERE O.Utente = offerta.Utente );

    COMMIT;
END
```

- **delete\_parent\_category:** Procedura che permette ad un amministratore di rimuovere la 'categoriaGenitore' da una categoria, con questa operazione in sostanza si alza il livello di una categoria da sottocategoria a macrocategoria. Poiché più amministratori potrebbero voler accedere alle stesse righe della tabella si sceglie il livello di isolamento Repeatable Read.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `delete_parent_category`( in var_name VARCHAR(45) )
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    SET TRANSACTION ISOLATION LEVEL REPEATABLE READ ;
    START TRANSACTION;

    UPDATE categoria
    SET categoriaGenitore = null
    WHERE nome = var_name;

    COMMIT;
END
```

- **delete\_category:** Procedura che permette di eliminare una categoria. Poiché è fondamentale che i dati siano confermati, vengono evitate Dirty Reads con il livello di isolamento Read Committed.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `delete_category`( in var_name VARCHAR(45) )
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    SET TRANSACTION ISOLATION LEVEL READ COMMITTED ;
    START TRANSACTION;

    DELETE FROM categoria WHERE nome = var_name;

    COMMIT;
END
```

- **insert\_parent\_category:** Procedura per l'inserimento di una categoria Genitore, le anomalie Dirty Reads sono accettabili in quanto non influenzano i dati, si sceglie Read Uncommitted.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `insert_parent_category`( in var_name VARCHAR(45) )
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
    START TRANSACTION;

    INSERT INTO categoria(nome) VALUES (var_name);

    COMMIT;
END
```



- **insert\_sub\_category**: Procedura per l'inserimento di una categoria specificando la categoria padre, si sceglie Repeatable Read per evitare che modifiche o eliminazioni della categoria genitore da parte di altre transazioni causino inconsistenze.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `insert_sub_category`( in var_name VARCHAR(45), in var_parent VARCHAR(45))
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
    START TRANSACTION;

    INSERT INTO categoria(nome, categoriaGenitore) VALUES (var_name, var_parent);

    COMMIT;
END
```

- **modify\_category\_procedure**: procedura per modificare il nome di una categoria. Per evitare inconsistenze, come sopra, si sceglie Repeatable Read.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `modify_category_procedure`( in var_old_name VARCHAR(45), in var_new_name VARCHAR(45) )
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
    START TRANSACTION;

    UPDATE categoria
    SET nome = var_new_name
    WHERE nome = var_old_name;

    COMMIT;
END
```

- **modify\_parent\_category\_procedure**: procedura per modificare la categoria genitore di una data categoria. Per la stessa motivazione delle precedenti si utilizza Repeatable Read.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `modify_parent_category_procedure`( in var_name VARCHAR(45), in var_parent VARCHAR(45) )
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
    START TRANSACTION;

    UPDATE categoria
    SET categoriaGenitore = var_parent
    WHERE nome = var_name;

    COMMIT;
END
```

- **show\_categories:** Procedura utilizzata dagli amministratori per visualizzare le categorie. Poiché non si evidenziano anomalie rilevanti si utilizza Read Uncommitted.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `show_categories`()
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
    START TRANSACTION;

    SELECT nome, categoriaGenitore
    FROM categoria ;

    COMMIT;
END
```

- **show\_open\_auction:** Procedura per mostrare tutte le informazioni delle aste ancora in corso.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `show_open_auction`()
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
    START TRANSACTION;

    SELECT ID, Descrizione, Stato, `Base d'asta`, Dimensioni, `Inizio asta`, `Termine asta`, `Numero offerte`, `Valore max offerta`, Categoria
    FROM `oggetto in asta`
    WHERE Tipo = 'in corso'
    AND `Termine asta` > CURRENT_DATE()
    UNION
    SELECT ID, Descrizione, Stato, `Base d'asta`, Dimensioni, `Inizio asta`, `Termine asta`, `Numero offerte`, `Valore max offerta`, Categoria
    FROM `oggetto in asta`
    WHERE Tipo = 'in corso'
    AND `Termine asta` = CURRENT_DATE()
    AND `Inizio asta` - CURRENT_TIME > 0;

    COMMIT;
END
```

- **make\_automatic\_offer:** Procedura per l'inserimento di un'offerta impostando la controfferta automatica. Inizialmente si effettuano i controlli per il corretto inserimento dell'offerta, viene quindi inserita una nuova offerta ed aggiornati i valori riguardanti numero offerte e valore massimo dell'oggetto in asta. Successivamente si cerca un utente con la controfferta automatica più alta diversa dall'utente corrente e, se la sua controfferta è maggiore dell'offerta corrente incrementata di 0.5, registra una nuova offerta. Come livello si è scelto Serializable per garantire che le offerte e controfferte siano gestite in modo coerente senza creare duplicati o incongruenze.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `make_automatic_offer`(in var_user VARCHAR(16), in var_amount FLOAT, in var_object VARCHAR(10), in var_controfferta FLOAT)
BEGIN
    DECLARE var_offer_date DATE;
    DECLARE var_offer_time TIME;
    DECLARE var_card_expire_date VARCHAR(5);
    DECLARE var_curr_max_offer FLOAT;
    DECLARE var_object_type ENUM('in corso', 'venduto', 'non venduto');
    DECLARE var_offers_number INT;
    DECLARE var_new_offers_number INT;
    DECLARE var_finish_auction DATE;
    DECLARE var_start_auction DATETIME;
    DECLARE var_base_price FLOAT;
    -- CONTROFFERTA
    DECLARE var_user_controfferta VARCHAR(16);
    DECLARE var_amount_controfferta FLOAT;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
```

```

BEGIN
    ROLLBACK;
    RESIGNAL;
END;

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
START TRANSACTION;

SELECT 'Base d'asta', 'Numero offerte', 'Valore max offerta', 'Termine asta', 'Inizio asta', Tipo
INTO var_base_price, var_offers_number, var_curr_max_offer, var_finish_auction, var_start_auction, var_object_type
FROM `oggetto in asta`
WHERE ID = var_object;

IF var_amount % 0.5 <> 0 THEN
    SIGNAL SQLSTATE '45010' SET MESSAGE_TEXT = "Offer amount not valid";
END IF;

IF var_controfferta < var_amount THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "Automatic offer amount can't be less than current offer ";
END IF;

IF var_amount <= var_curr_max_offer THEN
    SIGNAL SQLSTATE '45011' SET MESSAGE_TEXT = "Offer amount too low";
END IF;

IF var_offers_number = 0 THEN
    IF var_amount < var_base_price THEN
        SIGNAL SQLSTATE '45012' SET MESSAGE_TEXT = "Offer amount less than start price ";
    END IF;
END IF;

IF var_object_type <> 'in corso' THEN
    SIGNAL SQLSTATE '45013' SET MESSAGE_TEXT = "Auction finished";
END IF;

IF var_finish_auction < CURRENT_DATE() THEN
    SIGNAL SQLSTATE '45013' SET MESSAGE_TEXT = "Auction finished";
END IF;

IF var_finish_auction = CURRENT_DATE() AND TIME(var_start_auction) < CURRENT_TIME() THEN
    SIGNAL SQLSTATE '45013' SET MESSAGE_TEXT = "Auction finished";
END IF;

SELECT Scadenza INTO var_card_expire_date
FROM `carta di credito` JOIN Utente ON `carta di credito`.Numero = Utente.`Carta di Credito`
WHERE CF = var_user;
-- il pagamento avviene alla fine dell'asta, quindi è importante che la carta non scada prima della fine di essa
IF STR_TO_DATE(CONCAT('01-',var_card_expire_date), '%d-%m-%y') < var_finish_auction THEN
    SIGNAL SQLSTATE '45007' SET MESSAGE_TEXT = "Expired Credit Card ";
END IF;

SET var_offer_date = CURRENT_DATE();
SET var_offer_time = CURRENT_TIME();

SELECT 'Numero offerte' INTO var_offers_number
FROM `oggetto in asta`
WHERE ID = var_object;

SET var_new_offers_number = var_offers_number + 1;
-- inserimento offerta
INSERT INTO offerta( Utente, Importo, `Oggetto in asta`, `Data`, Ora, `importo massimo automatico` )
VALUES ( var_user, var_amount, var_object, var_offer_date, var_offer_time, var_controfferta );
-- aggiorno numero offerte e max offerta
UPDATE `oggetto in asta`
SET `Numero offerte` = var_new_offers_number, `Valore max offerta` = var_amount
WHERE ID = var_object;

-- cerco utente controfferta automatica
SELECT Utente, `Importo massimo automatico`
INTO var_user_controfferta, var_amount_controfferta
FROM offerta
WHERE `oggetto in asta` = var_object
AND `importo massimo automatico` IS NOT NULL
AND Utente <> var_user
GROUP BY Utente, `Importo massimo automatico`
HAVING MAX(`Importo massimo automatico`) > var_controfferta
ORDER BY `importo massimo automatico` DESC
LIMIT 1;

IF var_amount_controfferta IS NOT NULL AND var_amount_controfferta >= var_amount + 0.5 THEN

    INSERT INTO offerta( Utente, Importo, `Oggetto in asta`, `Data`, Ora, `importo massimo automatico`)
    VALUES (var_user_controfferta, var_amount + 0.5, var_object, current_date(), current_time());

    UPDATE `oggetto in asta`
    SET `Numero offerte` = `Numero offerte` + 1, `Valore max offerta` = var_amount + 0.5
    WHERE ID = var_object;

END IF;

COMMIT;
END

```

- **make\_offer:** Procedura per l'inserimento di un'offerta. Inizialmente si effettuano i controlli per il corretto inserimento dell'offerta, viene quindi inserita una nuova offerta ed aggiornati i valori riguardanti numero offerte e valore massimo dell'oggetto in asta. Successivamente si cerca un utente con la controfferta automatica più alta diversa dall'utente corrente e, se la sua controfferta è maggiore dell'offerta corrente incrementata di 0.5, registra una nuova offerta. Come livello si è scelto Serializable per garantire che le offerte e controfferte siano gestite in modo coerente senza creare duplicati o incongruenze.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `make_offer`(in var_user VARCHAR(16), in var_amount FLOAT, in var_object VARCHAR(10))
BEGIN
    DECLARE var_offer_date DATE;
    DECLARE var_offer_time TIME;
    DECLARE var_card_expire_date VARCHAR(5);
    DECLARE var_curr_max_offer FLOAT;
    DECLARE var_object_type ENUM('in corso', 'venduto', 'non venduto');
    DECLARE var_offers_number INT;
    DECLARE var_new_offers_number INT;
    DECLARE var_finish_auction DATE;
    DECLARE var_start_auction DATETIME;
    DECLARE var_base_price FLOAT;
    -- CONTROFFERTA
    DECLARE var_user_controfferta VARCHAR(16);
    DECLARE var_amount_controfferta FLOAT;
    DECLARE EXIT HANDLER FOR SQLWARNING
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
    START TRANSACTION;

    SELECT `Base d'asta`, `Numero offerte`, `Valore max offerta`, `Termine asta`, `Inizio asta`, Tipo
    INTO var_base_price, var_offers_number, var_curr_max_offer, var_finish_auction, var_start_auction, var_object_type
    FROM `oggetto in asta`
    WHERE ID = var_object;

    IF var_amount % 0.5 <> 0 THEN
        SIGNAL SQLSTATE '45010' SET MESSAGE_TEXT = "Offer amount not valid";
    END IF;

    IF var_amount <= var_curr_max_offer THEN
        SIGNAL SQLSTATE '45011' SET MESSAGE_TEXT = "Offer amount too low";
    END IF;

    IF var_offers_number = 0 THEN
        IF var_amount < var_base_price THEN
            SIGNAL SQLSTATE '45012' SET MESSAGE_TEXT = "Offer amount less than start price ";
        END IF;
    END IF;

    IF var_object_type <> 'in corso' THEN
        SIGNAL SQLSTATE '45013' SET MESSAGE_TEXT = "Auction finished";
    END IF;

    IF var_finish_auction < CURRENT_DATE() THEN
        SIGNAL SQLSTATE '45013' SET MESSAGE_TEXT = "Auction finished";
    END IF;

    IF var_finish_auction = CURRENT_DATE() AND TIME(var_start_auction) < CURRENT_TIME() THEN
        SIGNAL SQLSTATE '45013' SET MESSAGE_TEXT = "Auction finished";
    END IF;

    SELECT Scadenza INTO var_card_expire_date
    FROM `carta di credito` JOIN Utente ON `carta di credito`.Numero = Utente.`Carta di Credito`
    WHERE CF = var_user;
    -- il pagamento avviene alla fine dell'asta, quindi è importante che la carta non scada prima della fine di essa
    IF STR_TO_DATE(CONCAT('01-', var_card_expire_date), '%d-%m-%y') < var_finish_auction THEN
        SIGNAL SQLSTATE '45007' SET MESSAGE_TEXT = "Expired Credit Card ";
    END IF;

    SET var_offer_date = CURRENT_DATE();
    SET var_offer_time = CURRENT_TIME();

    SELECT `Numero offerte` INTO var_offers_number
    FROM `oggetto in asta`
    WHERE ID = var_object;
```

```
SET var_new_offers_number = var_offers_number + 1;
-- inserimento offerta
INSERT INTO offerta( Utente, Importo, 'Oggetto in asta', 'Data', Ora )
VALUES ( var_user, var_amount, var_object, var_offer_date, var_offer_time );
-- aggiorno numero offerte e max offerta
UPDATE 'oggetto in asta'
SET 'Numero offerte' = var_new_offers_number, 'Valore max offerta' = var_amount
WHERE ID = var_object;
-- cerco utente controfferta automatica
SELECT Utente, 'Importo massimo automatico'
INTO var_user_controfferta, var_amount_controfferta
FROM offerta
WHERE 'oggetto in asta' = var_object
AND 'importo massimo automatico' is not null
AND Utente <> var_user
GROUP BY Utente, 'Importo massimo automatico'
HAVING MAX('importo massimo automatico')
ORDER BY 'importo massimo automatico' DESC
LIMIT 1;
IF var_amount_controfferta is not null AND var_amount_controfferta >= var_amount + 0.5 then

    INSERT INTO offerta( Utente, Importo, 'Oggetto in asta', 'Data', Ora, 'importo massimo automatico')
    VALUES (var_user_controfferta, var_amount + 0.5, var_object, current_date(), current_time(), var_amount_controfferta);

    UPDATE 'oggetto in asta'
    SET 'Numero offerte' = 'Numero offerte' + 1, 'Valore max offerta' = var_amount + 0.5
    WHERE ID = var_object;

END IF;

COMMIT;

END
```