



Basi di Dati

Progetto A.A. 2024/2025

## SISTEMA DI ASTE ONLINE

0311206

Matteo Trossi

### Indice

1. Descrizione del Minimondo.....	2
2. Analisi dei Requisiti .....	4
3. Progettazione concettuale.....	9
4. Progettazione logica .....	21
5. Progettazione fisica .....	38

## 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 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  $I + 0,50\text{€}$ .

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

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

## 2. Analisi dei Requisiti

### Identificazione dei termini ambigui e correzioni possibili

Linea	Termine	Nuovo termine	Motivo correzione
2,9,13	Amministratori	Utenti Amministratori	Si vuole rendere più chiaro che sono una specializzazione del concetto generale di utente, andando a rendere esplicita la generalizzazione tra essi.
16	Amministratori del sistema	Utenti Amministratori	Si vuole unificare il sinonimo con un termine univoco e preciso per identificare il tipo di utente, riducendo le ambiguità.
4,9,12,22,33	Oggetto	Articolo	Viene sostituito il termine generico con uno più preciso e chiaro nel contesto delle aste.
2,9,16,37,38	Oggetti	Articoli	Si uniforma il termine per lo stesso motivo della voce precedente.
3,17	Utenti del sistema	Utenti Base	Si vuole chiarire che sono utenti normali, distinguendosi dagli utenti amministratori.
18,22,37	Utenti	Utenti Base	Termine che presenta degli omonimi, si vuole distinguere un utente generico, senza specificare il tipo esatto, dall'Utente Base del sistema, uniformando tutti i termini ambigui.
24,25,26	Utente	Utente Base	Si vuole uniformare il termine, come nella voce precedente, specificando il tipo esatto.
4	Offerta maggiore	Offerta massima	Si vuole uniformare il termine che presenta dei sinonimi, in modo da utilizzare un sinonimo più adatto nel contesto delle aste.
8	Indirizzo cui consegnare eventuali oggetti acquistati	Indirizzo di consegna	Si semplifica la formulazione, rendendo lo stesso concetto in modo più semplice e conciso.
10	Codice alfanumerico univoco	Identificatore	Viene utilizzato un termine tecnico più chiaro nel contesto delle basi di dati.
10	Stato	Condizioni dell'articolo	Si vogliono evitare ambiguità con lo stato dell'asta.
13	Decidere la durata dell'asta	Impostare la durata dell'asta	Viene sostituito un termine generico con uno più tecnico che esprime l'azione fisica effettuata dall'utente.
15	Titolario gerarchico	Struttura gerarchica delle categorie	Si vuole rendere più chiara e comprensibile la rappresentazione delle categorie di articoli.
17	Aste aperte	Aste attive	Si unifica il termine che presenta sinonimi con un termine più preciso e tecnico.
38	Asta in corso	Asta attiva	Si vuole uniformare il termine, come nella voce precedente, rendendolo più adatto al contesto tecnico.

**Specifica disambiguata**

Una casa d'aste intende realizzare un sistema online di aste. Il sistema deve consentire agli Utenti Amministratori la gestione degli articoli che si vogliono pubblicare e tutto il ciclo di vita delle aste. Gli Utenti Base, previa registrazione, hanno la possibilità di fare offerte su un qualsiasi articolo. Al termine dell'asta, l'offerta massima 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 di consegna.

Gli Utenti Amministratori gestiscono l'inserimento degli articoli. Ogni articolo è caratterizzato da un identificatore, una descrizione, le condizioni dell'articolo (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 articolo nel sistema, gli Utenti Amministratori possono impostare 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 una struttura gerarchica delle categorie, organizzata su un massimo di tre livelli. La gestione delle categorie degli articoli afferisce sempre agli Utenti Amministratori. Gli Utenti Base possono visualizzare in qualsiasi momento tutte le aste attive. Quando un'asta viene visualizzata, gli Utenti Base 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. Gli Utenti Base non possono però visualizzare chi è che ha effettuato l'offerta massima.

Dato un articolo in asta, gli Utenti Base 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 Base che ha attualmente piazzato l'offerta massima, può sfruttare la funzionalità di "controfferta automatica". Tale funzionalità permette all'Utente Base di indicare un importo massimo con cui si intende rilanciare l'offerta, qualora un altro Utente Base 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 articolo, 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 transazioni automatiche generate dal sistema di controfferta automatica devono essere registrate nel sistema.

Gli Utenti Base, in ogni momento, possono visualizzare l'elenco degli articoli aggiudicati e l'elenco degli articoli per i quali è presente un'asta attiva cui hanno fatto almeno un'offerta.

## Glossario dei Termini

Termine	Descrizione	Sinonimi	Collegamenti
Utente Amministratore	Gestisce gli articoli da pubblicare, includendo anche la gestione delle categorie, e il ciclo di vita delle aste associate ad essi.	Amministratore	Articolo, Asta, Categoria
Utente Base	Fa uso del sistema per partecipare alle aste attive e fare delle offerte sugli articoli.	Utente, Utente del sistema	Articolo, Asta, Offerta, Controfferta automatica
Articolo	Oggetto fisico messo all'asta dagli Utenti Amministratori, su cui qualunque Utente Base può fare un'offerta se interessato ad acquistarlo.	Oggetto	Utente Amministratore, Utente Base, Asta, Categoria, Offerta
Offerta	Somma di denaro che un Utente Base intende offrire per un articolo di interesse, rilanciando sull'offerta massima attuale per vincere l'asta e ottenere l'articolo.	Puntata	Utente Base, Articolo, Asta
Asta	Processo di vendita competitiva tra Utenti Base, i quali cercano di aggiudicarsi un articolo di interesse comune effettuando la migliore offerta su quella massima attuale.		Utente Amministratore, Utente Base, Articolo, Offerta, Controfferta automatica
Categoria	Raggruppamento basato su un tema specifico che permette di classificare gli articoli. La gestione delle categorie è affidata agli Utenti	Tipologia, Tipo	Utente Amministratore, Articolo, Asta

	Amministratori.		
Controfferta automatica	Sistema automatico di rilancio su una specifica asta.	Puntata automatica, Rilancio, Rilancio automatico	Utente Base, Articolo, Asta

## Raggruppamento dei requisiti in insiemi omogenei

### Frasi di carattere generale

Una casa d'aste intende realizzare un sistema online di aste.

### Frasi relative a Utente Amministratore

Il sistema deve consentire agli Utenti Amministratori la gestione degli articoli che si vogliono pubblicare e tutto il ciclo di vita delle aste.

Gli Utenti Amministratori gestiscono l'inserimento degli articolo.

Quando viene inserito un nuovo articolo nel sistema, gli Utenti Amministratori possono impostare la durata dell'asta, da un minimo di un giorno ad un massimo di sette giorni.

La gestione delle categorie degli articoli afferisce sempre agli Utenti Amministratori.

### Frasi relative a Utente Base

Gli Utenti Base, previa registrazione, hanno la possibilità di fare offerte su un qualsiasi articolo.

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 di consegna.

Gli Utenti Base possono visualizzare in qualsiasi momento tutte le aste attive.

Quando un'asta viene visualizzata, gli Utenti Base 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 articolo in asta, gli Utenti Base possono fare un'offerta, maggiore del valore attuale di offerta.

[Un] Utente Base che ha attualmente piazzato l'offerta massima, può sfruttare la funzionalità di "controfferta automatica".

Tale funzionalità permette all'Utente Base di indicare un importo massimo con cui si intende rilanciare l'offerta, qualora un altro Utente Base faccia un'offerta maggiore.

Gli Utenti Base [...] possono visualizzare l'elenco degli articoli aggiudicati e l'elenco degli articoli per i quali è presente un'asta attiva cui hanno fatto almeno un'offerta.

### Frasi relative all'Articolo

Il sistema deve consentire agli Utenti Amministratori la gestione degli articoli che si vogliono pubblicare e tutto il ciclo di vita delle aste.

Gli Utenti Base, previa registrazione, hanno la possibilità di fare offerte su un qualsiasi articolo.

Gli Utenti Amministratori gestiscono l'inserimento degli articolo.

Ogni articolo è caratterizzato da un identificatore, una descrizione, le condizioni dell'articolo (ad esempio "come nuovo", "in buone condizioni", "non funzionante", ecc.), da un prezzo di base d'asta, e da una descrizione delle dimensioni.

Il sistema tiene traccia, per ogni articolo, di tutte le offerte che sono state fatte e dell'istante temporale in cui queste sono state inserite nel sistema.

Gli Utenti Base, in ogni momento, possono visualizzare l'elenco degli articoli aggiudicati e l'elenco

degli articoli per i quali è presente un'asta attiva cui hanno fatto almeno un'offerta.

#### **Frasi relative all'Offerta**

Gli Utenti Base, previa registrazione, hanno la possibilità di fare offerte su un qualsiasi articolo.

Al termine dell'asta, l'offerta massima sarà quella che avrà vinto l'asta.

Dato un articolo in asta, gli Utenti Base possono fare un'offerta, maggiore del valore attuale di offerta.

La granularità di incremento delle offerte è di multipli di 50 centesimi di euro.

Il sistema tiene traccia, per ogni articolo, di tutte le offerte che sono state fatte e dell'istante temporale in cui queste sono state inserite nel sistema.

#### **Frasi relative all'Asta**

Il sistema deve consentire agli Utenti Amministratori la gestione degli articoli [...] e tutto il ciclo di vita delle aste.

Al termine dell'asta, l'offerta massima sarà quella che avrà vinto l'asta.

Quando viene inserito un nuovo articolo nel sistema, gli Utenti Amministratori possono impostare la durata dell'asta, da un minimo di un giorno ad un massimo di sette giorni.

Inoltre, a ciascuna asta viene associata una categoria.

Gli Utenti Base possono visualizzare in qualsiasi momento tutte le aste attive.

Quando un'asta viene visualizzata, gli Utenti Base ottengono tutte le informazioni legate allo stato attuale della stessa [...].

#### **Frasi relative alla Categoria**

[A] Ciascuna asta viene associata una categoria.

Le categorie appartengono ad una struttura gerarchica delle categorie, organizzata su un massimo di tre livelli.

La gestione delle categorie degli articoli afferisce sempre agli Utenti Amministratori.

#### **Frasi relative alla Controfferta automatica**

[Un] Utente Base che ha attualmente piazzato l'offerta massima, può sfruttare la funzionalità di "controfferta automatica".

Tale funzionalità permette all'Utente Base di indicare un importo massimo con cui si intende rilanciare l'offerta [...].

L'utente A indica un importo I con cui vuole rilanciare [...]. 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 transazioni automatiche generate dal sistema di controfferta automatica devono essere registrate nel sistema.

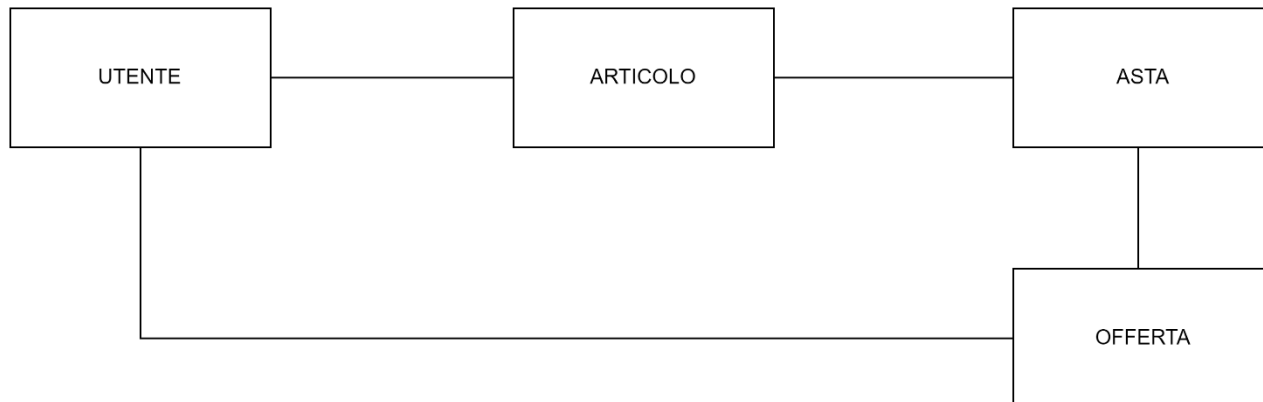


### 3. Progettazione concettuale

#### Costruzione dello schema E-R

Per la costruzione del diagramma E-R si adotta un approccio misto, combinando un metodo top-down e un metodo inside-out (una variante del bottom-up). Inizialmente, si definisce uno schema generale identificando i concetti astratti fondamentali della specifica di progetto, che verranno successivamente affinati. Parallelamente, si utilizza un approccio inside-out per espandere progressivamente lo schema, partendo dai concetti chiave individuati e sviluppandolo fino a includere tutti gli elementi necessari. Si è deciso di adottare un approccio misto per la strutturazione dello schema E-R perché ci consente di combinare i vantaggi offerti singolarmente dagli approcci top-down e inside-out.

Partendo con l'approccio top-down, sono stati individuati i principali concetti essenziali della specifica di progetto, in modo tale da mantenere una visione globale dei concetti che dovranno costituire il sistema finale:

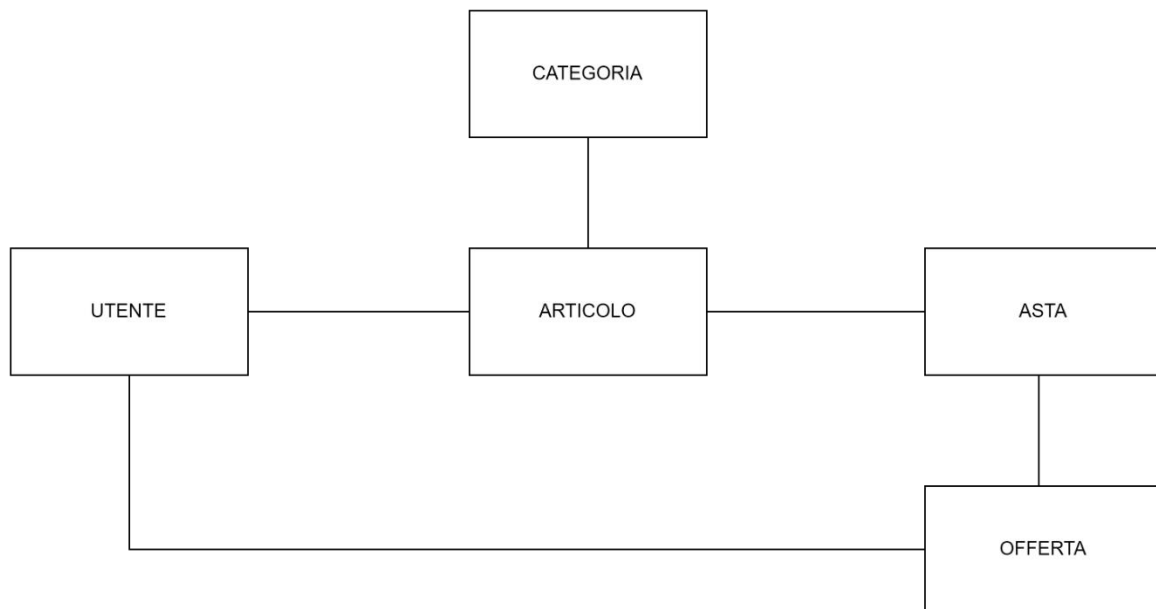


Si individuano le seguenti entità fondamentali:

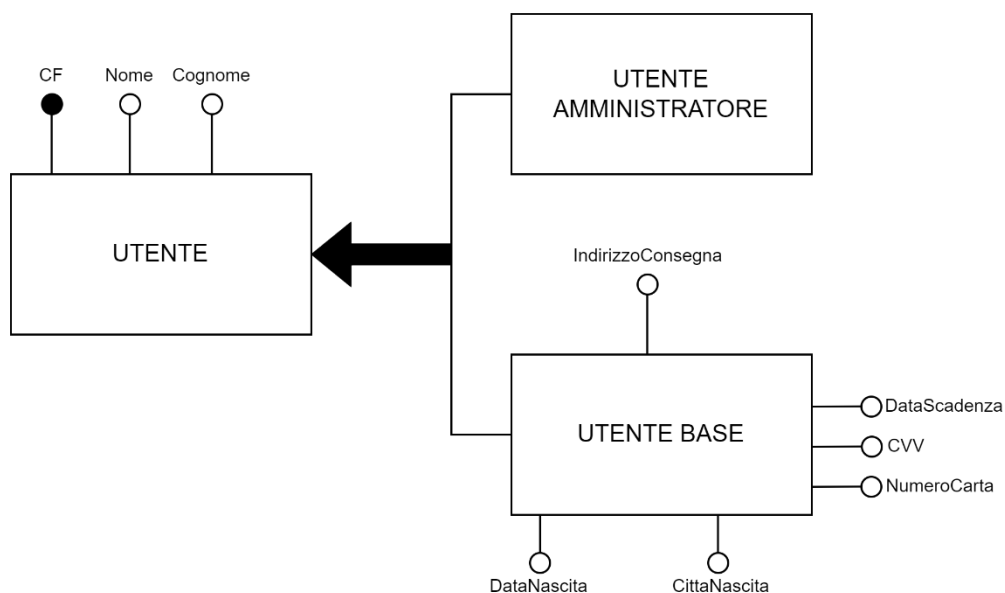
- **Utente:** l'attore che partecipa alle aste o che gestisce la pubblicazione degli articoli e di conseguenza le aste associate ad essi.
- **Articolo:** entità ancora astratta che rappresenta l'oggetto messo all'asta.
- **Asta:** entità che rappresenta la competizione per acquistare un articolo.
- **Offerta:** entità che rappresenta in modo ancora astratto la somma di denaro proposta da un utente per aggiudicarsi la vittoria in un'asta.

Si osserva come, in questo schema scheletrico, punto di partenza per un'espansione progressiva secondo l'approccio *inside-out*, risulti assente l'entità *Categoria*, che rappresenta invece una componente fondamentale del progetto.

Si aggiunge l'entità *Categoria* e le relative associazioni astratte che dovranno essere affinate in seguito:



Proseguendo sempre con l'approccio top-down, si nota come l'utente abbia più ruoli completamente distinti, proprio come specificato nel testo del progetto. Si procede quindi alla trasformazione di un'entità in una gerarchia di generalizzazione:

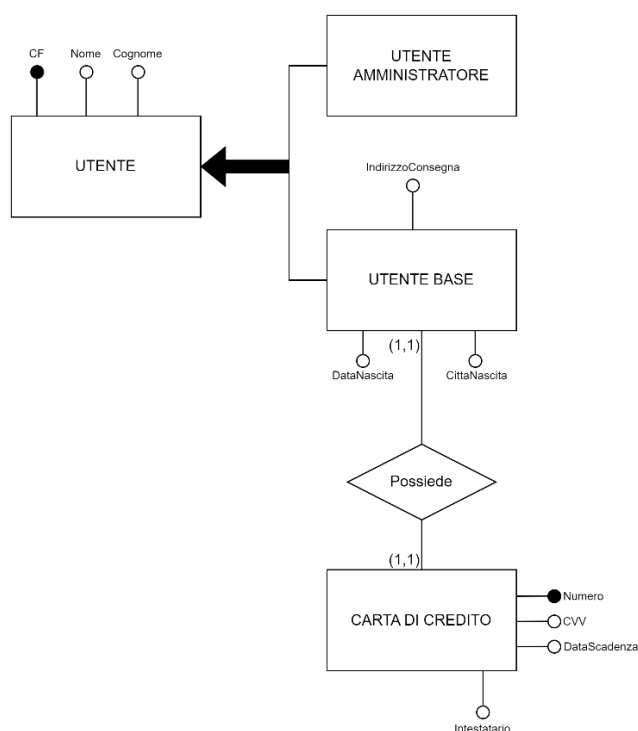


Una prima realizzazione di questa gerarchia di generalizzazione presenta *Utente* come entità padre, pensata per rappresentare in modo generico gli utenti ed evitare ridondanze, dato che senza specializzazione avremmo dovuto ripetere gli attributi “CF”, “Nome” e “Cognome”, che sono comuni tra i figli. Invece, come entità figlie, che specializzano *Utente*, abbiamo *Utente Base* e *Utente Amministratore*. Si è scelto di separare i ruoli in due entità separate perché hanno attributi e comportamenti molto diversi. Infatti, l’*Utente Amministratore* non aggiunge nuovi attributi all’*Utente* generico, mentre l’*Utente Base* ha bisogno di altri attributi, come l’Indirizzo di Consegna e le informazioni sul metodo di pagamento, che sono necessari per effettuare offerte per partecipare alle aste e aggiudicarsi gli articoli.

Si è deciso, inoltre, di utilizzare una *generalizzazione totale* sull’entità *Utente* per enfatizzare il diverso ruolo che hanno nel sistema questi due tipi di utenti. Non viene utilizzata una generalizzazione parziale poiché nel sistema esistono solo queste due tipologie di utenti.

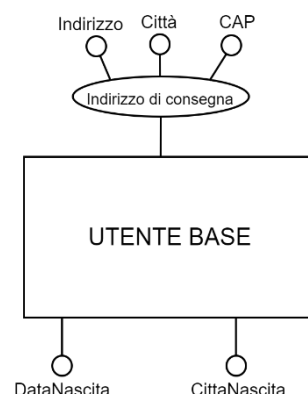
Successivamente, si è proceduto con un’ulteriore perfezionamento dell’entità *Utente Base*, adottando questa volta un approccio inside-out, introducendo l’entità *Carta di credito* tramite reificazione. Questa scelta è derivata dalla necessità di separare concetti diversi, evitare ridondanze e garantire una flessibilità maggiore in caso di aggiunta di ulteriori metodi di pagamento in futuro, evitando quindi di appesantire l’entità *Utente Base*.

La rappresentazione che ne è derivata è la seguente:

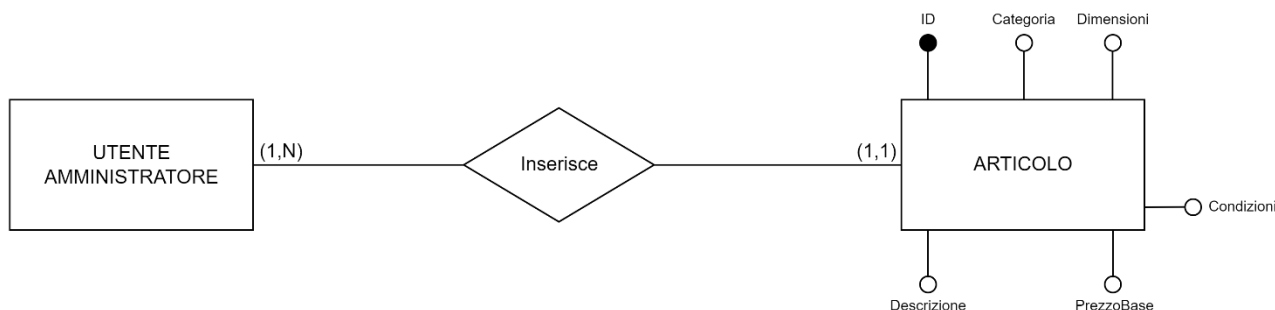


In questo modo una Carta di Credito è associata ad un solo Utente Base, identificandola con il suo Numero di carta univoco, e anche l'Utente Base possiede per l'attuale progettazione una sola Carta di Credito da utilizzare come metodo di pagamento nel sistema.

Proseguendo, si è deciso di affinare ulteriormente l'attributo *Indirizzo di consegna* di Utente Base andando a renderlo un *attributo composto*, in modo da rispettare la 1NF. Infatti, l'attributo Indirizzo di consegna non conteneva solo valori atomici, perché scomponibile in Indirizzo, Città e CAP; per questo motivo è stato suddiviso in ulteriori tre attributi associati all'attributo composto.

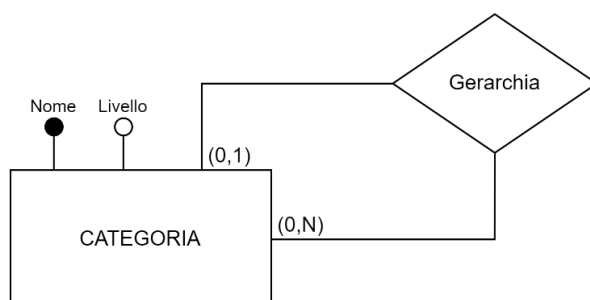


Proseguendo, è stata definita l'associazione principale tra *Utente Amministratore* e *Articolo*, poiché esso è in grado di inserire più Articoli differenti nel sistema:



L'attributo "*Dimensioni*" dell'entità Articolo non viene trasformato in attributo composto perché andremo ad intendere le dimensioni in modo atomico, rappresentandole come una singola stringa. Questa scelta è derivata dal fatto che non sarà necessario utilizzare questo dato in modo attivo, ma sarà mostrato agli utenti come informazione statica dell'articolo messo all'asta.

Proseguendo, si è deciso di modellare l'entità *Categoria* in questo modo:



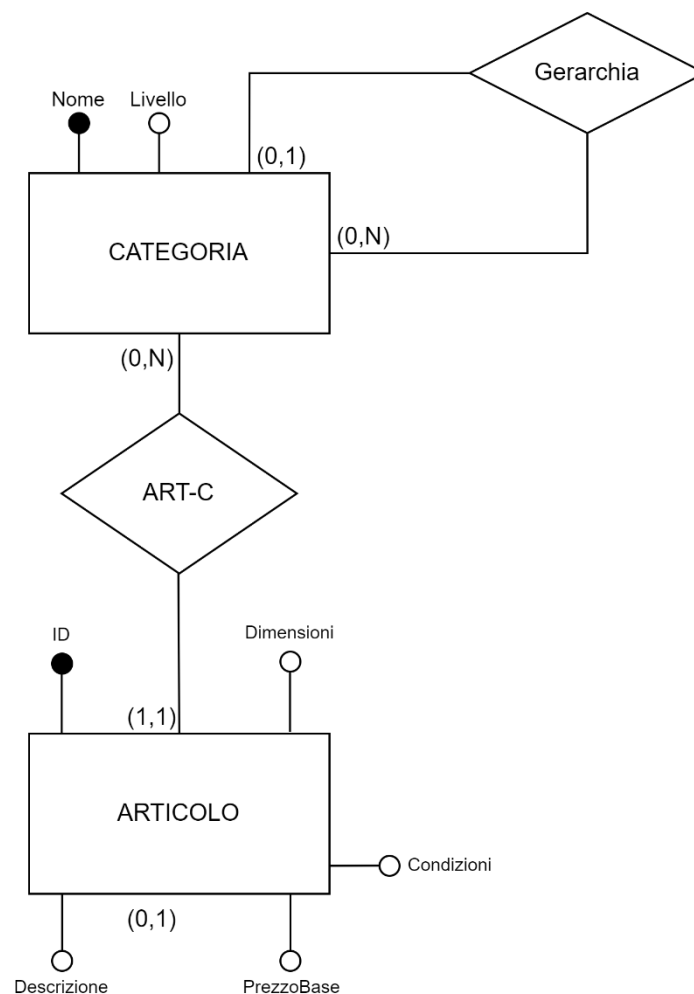
In questo modello, una *Categoria* è identificata univocamente dal proprio Nome. Il campo Livello viene impiegato per garantire vincoli di integrità e coerenza aziendale, limitando la profondità della gerarchia a un massimo di tre livelli. Questa struttura consente l'implementazione di filtri efficaci e scalabili nella navigazione e nella ricerca delle aste.

Di conseguenza, saranno definite apposite regole aziendali che imporranno:

- un limite massimo di tre livelli nella gerarchia;
- il divieto per una categoria di livello  $X$  di essere figlia di una categoria di livello  $Y$ , se  $X > Y$ .

Le cardinalità dell'associazione *Gerarchia* riflettono il fatto che una categoria può avere al più un'unica categoria padre, ma può possedere molteplici categorie figlie.

Viene creata anche l'associazione *ART-C* tra Categoria e Articolo:

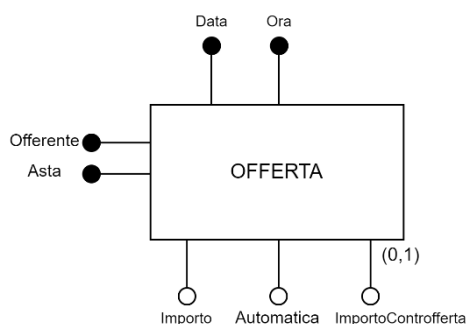
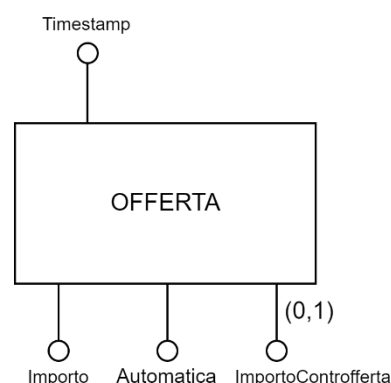


In questo modo viene mostrato come una *Categoria* possa avere più *Articoli* assegnati e come un singolo Articolo abbia assegnata una sola Categoria.

Inoltre, si è scelto di creare un'entità *Offerta*, in modo da poter rappresentare ogni proposta economica effettuata da un utente durante un'asta. Essa è stata modellata con gli attributi *Timestamp* e *Importo*, come da specifica di progetto.

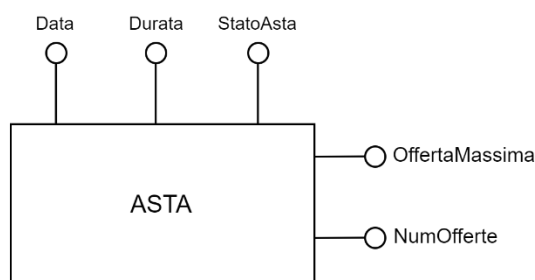
Successivamente, si è deciso di aggiungere un attributo che è chiamato *Automatica*, atto a differenziare le offerte manuali da quelle con importo di controfferta automatica impostato.

La gestione delle controfferte automatiche è implementata lato client, ma il sistema deve comunque registrare ogni offerta automatica come record a tutti gli effetti nel database, al pari di un'offerta manuale. Questo approccio consente di mantenere una cronologia completa delle offerte, garantendo trasparenza, tracciabilità e corretta determinazione del vincitore. Per questi motivi, si è resa necessaria la creazione dell'attributo opzionale *ImportoControfferta*, il quale consente di distinguere tra l'importo effettivamente offerto in quel momento (*Importo*) e la soglia massima preimpostata che abilita futuri rilanci automatici in risposta ad altre offerte. Tale attributo è stato reso opzionale poiché non necessario in caso di offerte manuali.



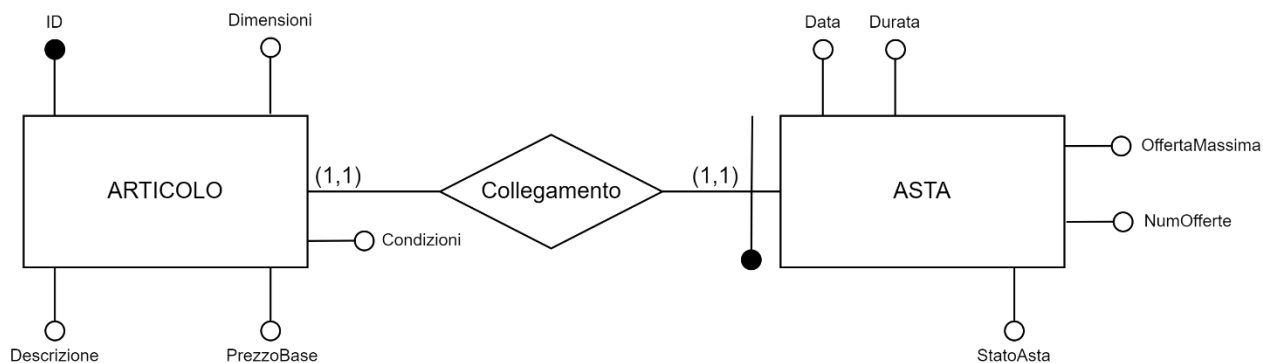
Per raffinare ulteriormente la progettazione, si è scelto di dividere l'attributo *Timestamp* in due attributi *Data* e *Ora*, andando a rispettare il principio di atomicità dei dati. Effettuando questa separazione si ottiene una gestione più flessibile e chiara delle informazioni temporali, potendo anche effettuare eventuali filtri futuri oppure operazioni di aggregazione per data e ora, consentendo un'analisi più accurata da parte degli Utenti Amministratori. Sono stati aggiunti come identificatori anche l'*Offerente* e l'*Asta*, essendo necessari per associare l'*Offerta* ad un contesto preciso.

Di conseguenza, si passa ora all'entità *Asta*, che rappresenta la parte di schema più importante essendo il nucleo di tutto il funzionamento dell'applicazione:

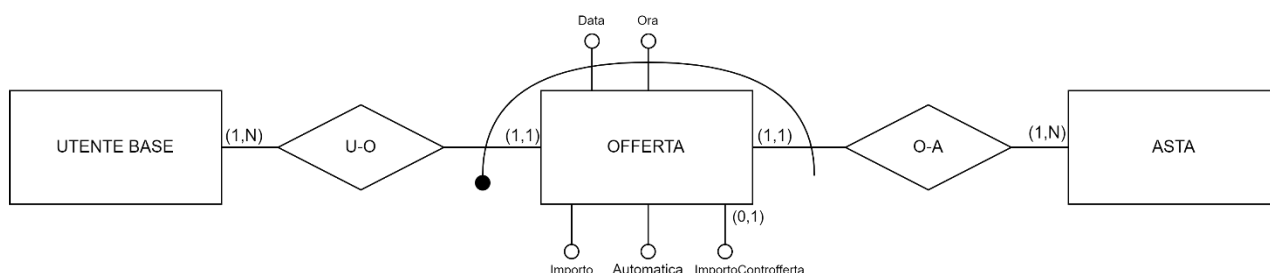


Come da specifica, sono stati aggiunti gli attributi di *Data*, rappresentate la data di creazione dell'asta, la *Durata*, che potrà variare da 1 a 7 giorni, l'*Offerta massima* e il *Numero Offerte*. L'*Offerta massima* rappresenta l'importo più alto offerto dagli utenti e il *Numero Offerte* è un valore numerico incrementato ad ogni offerta fatta, a partire dal valore iniziale 0 alla creazione dell'*Asta*. La *Durata* verrà gestita all'atto di inizializzazione di una nuova asta su un articolo inserito e ci sarà una regola aziendale che andrà a specificare l'intervallo di valori legali. Inoltre, l'attributo *StatoAsta* avrà il compito di specificare lo stato corrente di un'asta in uno specifico istante temporale, in particolare durante la visualizzazione dei dettagli di un'asta.

Dopo un'analisi più approfondita, si è deciso di modellare *Asta* come entità debole, strettamente dipendente dall'entità *Articolo*. L'associazione tra *Asta* e *Articolo* è di cardinalità uno-a-uno e l'*Asta* ha come identificatore univoco l'associazione con *Articolo*, avendo deciso di avere una singola asta per ogni articolo pubblicato dagli Utenti Amministratori.



Proseguendo nella realizzazione del diagramma E-R, si realizzano le associazioni necessarie a collegare gli *Utenti Base* con le *Offerte* e le *Offerte* con le *Aste*. La modellazione è stata realizzata in questo modo:



Sono state introdotte due nuove associazioni con cardinalità uno-a-molti per rappresentare correttamente il fatto che un Utente Base può effettuare più Offerte nel tempo, e che una singola Asta può ricevere numerose offerte da utenti diversi.

Per garantire l'univocità di ciascuna Offerta, si è scelto di utilizzare come identificatore una combinazione di associazioni e attributi:

- l'associazione *Utente-Offerta*, che collega l'offerta all'utente che la effettua;
- l'associazione *Offerta-Asta*, che lega l'offerta all'asta di riferimento;
- gli attributi *Data* e *Ora*, che ne registrano il momento preciso.

Questa combinazione assicura che ogni offerta sia distinta non solo per il momento in cui viene effettuata, ma anche per l'utente che la realizza e per l'asta a cui si riferisce, evitando così ambiguità nel caso di offerte simultanee effettuate dallo stesso utente sulla stessa asta.

Per concludere la progettazione del diagramma E-R rimane soltanto da modellare l'associazione *Aggiudica*, la quale mette in relazione gli Utenti Base con gli Articoli da essi aggiudicati.

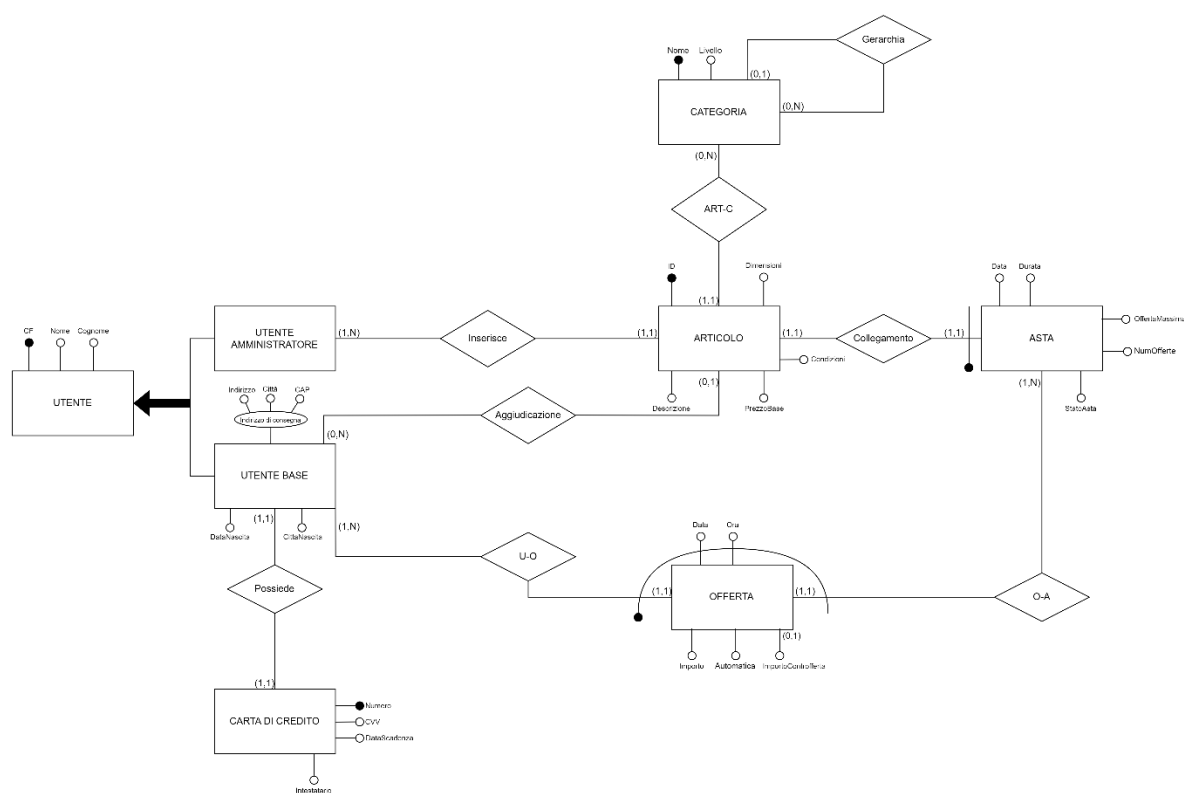




L'associazione *Aggiudicazione* presenta una cardinalità uno-a-molti, poiché un Utente Base può aggiudicarsi molti articoli diversi vincendo numerose aste, mentre un singolo articolo può essere aggiudicato da un solo Utente Base vincitore.

## Integrazione finale

Prima di procedere con la fase di progettazione logica e la conseguente ristrutturazione dello schema concettuale prodotto in questa fase, è necessario verificare la presenza di eventuali conflitti sui nomi utilizzati e di altri conflitti strutturali andando a procedere con la loro risoluzione. Lo schema integrato è il seguente:



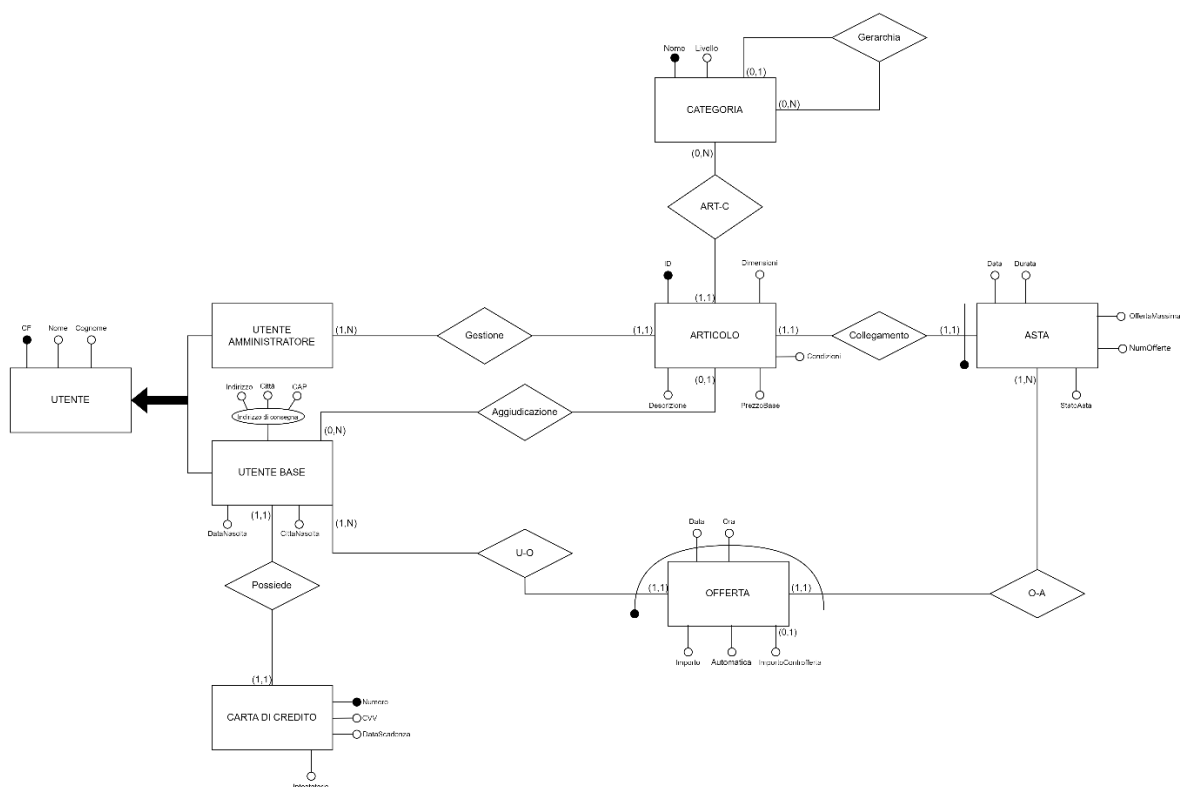
Si procede ora all'analisi di qualità dello schema prodotto andando a verificare: *correttezza*, *completezza*, *minimalità* e *leggibilità*:

- **correttezza:** lo schema risulta corretto sia sintatticamente che semanticamente, sono stati infatti rispettate tutte le regole principali di realizzazione e i vari costrutti adottati sono stati utilizzati nel modo corretto. Tuttavia, è stata effettuata una modifica sulla cardinalità di *Aggiudicazione* da (1,1) a (0,1) da *Articolo* verso *Utente Base*, perché un *Articolo* viene

aggiudicato da un singolo *Utente Base*, ma se l'*Asta* non si è ancora conclusa non è ancora stato effettivamente aggiudicato da qualcuno.

- **completezza:** è stata effettuata una rilettura completa della specifica disambiguata precedentemente, andando a testare la navigazione completa dello schema realizzato per controllare la possibilità di punti mancanti.
- **minimalità:** lo schema risulta nel complesso minimale, tranne per la presenza dell'attributo *Intestatario* su *Carta di Credito* e degli attributi *Offerta massima* e *Numero Offerte* su *Asta*, essendo calcolabili tramite le associazioni con *Utente Base* e *Offerta*. Tali ridondanze verranno discusse nella prossima fase di progettazione logica, in cui verranno analizzati i costi delle operazioni, andando a decidere se la loro presenza possa avere effetti positivi sulla complessità del sistema.
- **leggibilità:** la leggibilità risulta buona nel complesso dello schema. Non sono stati realizzati incroci tra le varie associazioni, gli attributi sono stati organizzati e rinominati in modo corretto per essere espliciti sul loro significato nello schema. Tuttavia, si è deciso di rinominare l'associazione *Inserisce* in *Gestione*, in quanto un amministratore non andrà soltanto ad inserire articoli da mettere all'asta, ma avrà le facoltà di gestione dell'intero ciclo di vita di un'asta.

Lo schema integrato finale risulta il seguente:



## Regole aziendali

- L'attributo *Durata* dell'entità *Asta* deve essere compreso tra un minimo di 1 giorno e un massimo di 7 giorni.
- Il *Prezzo base* deve essere maggiore del valore 0.
- Le *Categorie* devono appartenere ad una struttura gerarchica di massimo tre livelli.
- Ogni *Categoria* deve avere una *Categoria* padre, ad esclusione di quelle di livello 1.
- Se viene eliminata una *Categoria*, se presenti, le categorie figlie vengono eliminate automaticamente.
- Se una *Categoria* viene eliminata e vi sono *Articoli* associati, questi vengono automaticamente riassegnati a una *Categoria* di default predefinita.
- Un *Utente Base* non può offrire su una *Asta* terminata.
- Ogni *Offerta* ad un'*Asta* associata ad un certo *Articolo* deve avere *Importo* maggiore del valore dell'attributo *Prezzo base*.
- Un *Utente Base* non deve fare *Offerte* inferiori all'*Offerta* massima attuale + 0,50€.
- Il *tempo mancante alla chiusura dell'asta* si deve ottenere calcolando la differenza tra *DataFine*, calcolabile sommando la *Durata* alla *DataCreazione*, e la *DataAttuale* (oggi).

## Dizionario dei dati

Per le entità figlie sono riportati solo gli attributi e gli identificatori aggiuntivi rispetto all'entità padre.

Entità	Descrizione	Attributi	Identificatori
UTENTE	Entità base che rappresenta qualunque tipo di attore che opera nel sistema.	Nome, Cognome	CF
UTENTE BASE	Specializza <i>UTENTE</i> rappresentando il ruolo di utente normale nel sistema.	Indirizzo di consegna, DataNascita, CittàNascita	
UTENTE AMMINISTRATORE	Specializza <i>UTENTE</i> rappresentando il ruolo di amministratore di sistema.		

CARTA DI CREDITO	Racchiude i dati reali della carta di credito utilizzata dagli Utenti Base come metodo di pagamento nelle aste.	CVV DataScadenza, Intestatario	<i>Numero</i>
ARTICOLO	Oggetto fisico disponibile nel sistema per la vendita tramite messa all'asta.	Dimensioni, Descrizione, PrezzoBase, Condizioni	<i>ID</i>
OFFERTA	Somma di denaro proposta da un possibile acquirente che partecipa ad un'asta per l'acquisto di uno specifico articolo.	Importo, Automatica, ImportoControfferta	<i>U-O, Data, Ora, O-A</i>
CATEGORIA	Categorie in cui sono organizzati e raggruppati articoli e aste.	Livello	<i>Nome</i>
ASTA	Competizione di vendita di articoli basata su offerte.	Data, Durata, OffertaMassima, NumOfferte, StatoAsta	<i>Collegamento</i>

## 4. Progettazione logica

### Volume dei dati

Consideriamo di mantenere i dati relativi ad *Articoli*, *Aste* e *Offerte* per un periodo non superiore a 10 anni.

Inoltre, si suppone che vi siano 30 *Utenti Amministratori*, ciascuno dei quali inserisce circa 10 articoli da mettere all'asta quotidianamente.

Per ogni Utente Amministratore consideriamo una platea di 20 *Utenti Base*, portando il numero totale di *Utenti Base* a circa 600.

Si suppone anche che ciascun articolo messo all'asta riceva almeno 3 offerte, in modo da garantire un livello minimo di competizione per ciascuna asta attiva.

Per quanto riguarda gli articoli aggiudicati dagli *Utenti Base*, ci si aspetta che ciascuno di essi si aggiudichi circa 10 articoli al mese.

La *durata media* di un'asta è pari a 4 giorni, andando a definire come varia il numero di aste contemporaneamente in corso nel sistema.

Per quanto concerne la gestione delle categorie da poter assegnare ad ogni articolo, ogni *Utente Amministratore* inserisce 8 categorie, anche avendo la possibilità di inserire o rimuovere qualunque categoria nel sistema.

Supponiamo anche che ogni *Categoria* faccia parte di una gerarchia con mediamente 2 livelli e che siano il 30% di primo livello e la restante parte di secondo livello.

Si assume che la totalità, ossia il 100%, degli *Utenti Base*, iscritti regolarmente alla piattaforma, partecipino tutti attivamente ad almeno 2 aste contemporaneamente attive, facendo mediamente 3 *offerte per asta*. In particolare, circa il 15% delle *Offerte* totali in un'*Asta* viene generato dal sistema di controfferta automatica.

Infine, si assume anche che circa il 20% degli *Utenti Base* sia particolarmente attivo, andando ad effettuare un numero di offerte superiore alla media stabilita precedentemente.

Concetto nello schema	Tipo <sup>1</sup>	Volume atteso
Utente	E	630
Utente Amministratore	E	30
Utente Base	E	600
Categoria	E	240
Articolo	E	1.095.000

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

Asta	E	1.095.000
Offerta	E	15.768.000
Carta di credito	E	600
Gestione	R	1.095.000
Aggiudicazione	R	720.000
U-O	R	15.768.000
Possiede	R	600
ART-C	R	2.190.000
Gerarchia	R	168
Collegamento	R	1.095.000
O-A	R	15.768.000

Come già descritto all'inizio, i dati vengono mantenuti nel sistema per 10 anni dal loro inserimento; quindi, è prevista una cancellazione periodica su base mensile dei dati che hanno superato i 10 anni dalla loro data di inserimento. Tuttavia, in questa tabella del volume dei dati atteso non teniamo in considerazione tale cancellazione, ma ci limitiamo a stimare la quantità di dati gestita dal sistema nel tempo, senza considerare anche la possibilità reale che si iscrivano progressivamente nuovi Utenti e quindi aumenti il carico sul sistema. In questo modo stiamo definendo una stima di base sui dati attesi, che sarà però necessario monitorare costantemente per gestire l'impatto dovuto alla crescita del volume dei dati.

## Tavola delle operazioni

Nella seguente tabella vengono riportate tutte le operazioni non banali previste nel sistema. In particolare vengono definite tre classi di operazioni basate sui ruoli: L (*Login*), UA (*Utente Amministratore*), UB (*Utente Base*).

Cod.	Descrizione	Frequenza attesa
L1	Login	300/ora
UA1	Inserisci nuovo articolo	300/giorno
UA2	Inserisci nuova categoria	10/mese
UA3	Inizializza asta per articolo	300/giorno
UA4	Visualizza lo stato di un'asta	500/giorno
UA5	Visualizza le offerte associate ad un'asta	250/giorno
UB1	Visualizza le aste attive	250/ora
UB2	Visualizza le aste attive per categoria	100/ora
UB3	Visualizza le aste attive per utente amministratore	80/giorno
UB4	Visualizza tutti gli articoli aggiudicati da un Utente Base	200/giorno
UB5	Registra offerta su articolo in asta	150/ora
UB6	Visualizza elenco articoli associati ad un'asta attiva sui quali hai effettuato un'offerta	1500/giorno

## Costo delle operazioni

Per il calcolo del costo delle singole operazioni descritte nella precedente tabella si stima che un accesso in lettura abbia costo 1 (unitario), mentre un accesso in scrittura abbia costo 2 (doppio rispetto alla singola lettura).

Per ogni operazione descritta sommariamente nella tavola delle operazioni è utile riportare la *tavola degli accessi*, in modo da poter andare a stabilire i costi di ognuna. Per ogni tavola degli accessi è necessario riportare il concetto dello schema acceduto, il tipo di costrutto, il numero effettivo di accessi e il tipo di accesso (lettura/scrittura).

### Operazione L1 – Login

Concetto	Costrutto <sup>2</sup>	Accessi	Tipo <sup>3</sup>
Utente	E	1	L

Costo singola operazione: 1

Accessi totali: 300/ora

### Operazione UA1 – Inserisci nuovo articolo

Si considera la pubblicazione quotidiana di circa 10 nuovi articoli da parte di ogni Utente Amministratore.

Concetto	Costrutto	Accessi	Tipo
Articolo	E	1	S
Gestione	R	1	S

Costo singola operazione:  $2 \times 2 = 4$  accessi

Accessi totali:  $4 \times 300 = 1200$  accessi/giorno

### Operazione UA2 – Inserisci nuova categoria

Si considera la presenza di 240 categorie totali, di cui il 30% di primo livello e il restante 70% di secondo livello. L'aggiunta di una categoria di primo livello comporta soltanto la scrittura dell'entità Categoria. Invece, l'aggiunta di una categoria di secondo livello comporta la scrittura dell'entità

---

<sup>2</sup> E = entity, R = relation

<sup>3</sup> L = lettura, S = scrittura

Categoria e la scrittura dell'associazione Gerarchia per associarla con una Categoria di livello superiore.

Presentiamo la tabella dei costi in entrambi i casi:

Caso A) *aggiunta nuova categoria di primo livello*

Concetto	Costrutto	Accessi	Tipo
Categoria	E	1	S

Costo singola operazione:  $1 * 2 = 2$  accessi

Caso B) *aggiunta nuova categoria di secondo livello*

Concetto	Costrutto	Accessi	Tipo
Categoria	E	1	S
Gerarchia	R	1	S

Costo singola operazione:  $2 * 2 = 4$  accessi

Costo medio ponderato:  $(0.3 * 2) + (0.7 * 4) = 3.4$  accessi

Accessi totali:  $3.4 * 10 = 34$  accessi/mese

*Operazione UA3 – Inizializza asta per articolo*

*Si assume l'inserimento di 300 articoli al giorno, con conseguente inizializzazione di 300 aste al giorno.*

Concetto	Costrutto	Accessi	Tipo
Asta	E	1	S
Collegamento	R	1	S

Costo singola operazione:  $2 * 2 = 4$  accessi

Accessi totali:  $4 * 300 = 1200$  accessi/giorno

*Operazione UA4 – Visualizza lo stato di un'asta*

Concetto	Costrutto	Accessi	Tipo
Asta	E	1	L
Articolo	E	1	L
Collegamento	R	1	L
ART-C	R	1	L
Categoria	E	1	L

Costo singola operazione:  $5 * 1 = 5$  accessi

Accessi totali:  $5 * 500 = 2500$  accessi/giorno



*Operazione UA5 – Visualizza le offerte associate ad un’asta*

*Si assume mediamente 14 offerte per asta (3 minime + incremento per partecipazione attiva).*

Concetto	Costrutto	Accessi	Tipo
Asta	E	1	L
O-A	R	14	L
Offerta	E	14	L
U-O	R	14	L
Utente Base	E	14	L

Costo singola operazione:  $(1 + 14*4) = 57$  accessi

Accessi totali:  $57 * 250 = 14250$  accessi/giorno

*Operazione UB1 – Visualizza le aste attive*

*Si considera l’inserimento di circa 300 articoli al giorno, con la conseguente inizializzazione di 300 diverse aste per ciascun articolo pubblicato. Inoltre, si considera una durata media di 4 giorni per ogni asta inizializzata, avendo quindi una media di 1200 aste attive al giorno.*

Concetto	Costrutto	Accessi	Tipo
Asta	E	1200	L
Collegamento	R	1200	L
Articolo	E	1200	L

Costo singola operazione:  $1200 * 3 * 1 = 3600$  accessi

Accessi totali:  $3600 * 250 = 900.000$  accessi/ora

*Operazione UB2 – Visualizza le aste attive per categoria*

*Si assume la ricerca basata su Categorie foglia (circa 168) e che le aste attive quotidiane siano distribuite equamente tra queste categorie (circa 7 aste per categoria).*

Concetto	Costrutto	Accessi	Tipo
Categoria	E	1	L
ART-C	R	7	L
Articolo	E	7	L
Collegamento	R	7	L
Asta	E	7	L

Costo singola operazione:  $(1 + 7*4) = 29$  accessi

Accessi totali:  $29 * 100 = 2900$  accessi/ora

*Operazione UB3 – Visualizza le aste attive per utente amministratore*

*Si assume che ogni Utente Amministratore pubblichi quotidianamente 10 articoli e inicializzi 10 aste. Si assume anche che le aste abbiano durata media di circa 4 giorni.*

Concetto	Costrutto	Accessi	Tipo
Utente Amministratore	E	1	L
Gestione	R	40	L
Articolo	E	40	L
Collegamento	R	40	L
Asta	E	40	L

Costo singola operazione:  $(40 * 4 + 1) * 1 = 161$  accessi

Accessi totali:  $161 * 80 = 12.880$  accessi/giorno

*Operazione UB4 – Visualizza tutti gli articoli aggiudicati da un Utente Base*

*Si assume che un singolo Utente Base si aggiudichi circa 10 articoli al mese. Essendo questa un'operazione che viene fatta quotidianamente si assume anche che al momento dell'esecuzione un singolo Utente Base si sia aggiudicato circa la metà degli articoli mediamente aggiudicati in un intero mese.*

Concetto	Costrutto	Accessi	Tipo
Utente Base	E	1	L
Aggiudicazione	R	5	L
Articolo	E	5	L

Costo singola operazione:  $11 * 1 = 11$  accessi

Accessi totali:  $11 * 200 = 2200$  accessi/giorno

*Operazione UB5 – Registra offerta su articolo in asta*

*Si suppone la presenza degli attributi ridondanti Offerta massima e Numero Offerte sull'entità Asta. Questi attributi dovranno essere aggiornati ogni volta che un Utente Base aggiunge una nuova Offerta ad un'Asta; in particolare, il valore di Numero Offerte viene incrementato ad ogni nuova Offerte di una unità e il valore di Offerta massima viene aggiornato con l'ultima Offerta aggiunta per l'Asta che avrà importo sicuramente maggiore di tutte le precedenti.*

Concetto	Costrutto	Accessi	Tipo
----------	-----------	---------	------

Offerta	E	1	S
O-A	R	1	S
U-O	R	1	S
Asta	E	1	L
Asta	E	2	S

Costo singola operazione:  $(1 + 1 + 1 + 2) * 2 + 1 * 1 = 11$  accessi

Accessi totali:  $11 * 150 = 1650$  accessi/ora

*Operazione UB6* – Visualizza elenco articoli associati ad un’asta attiva sui quali hai effettuato un’offerta

*Si assume che ogni Utente Base partecipi attivamente a 2 aste al giorno; dunque, è necessario considerare anche le aste dei giorni precedenti che sono ancora in corso. Assumendo che un'asta abbia una durata media di 4 giorni, l'Utente Base potrebbe avere Offerte attive su Aste iniziate fino a 3 giorni prima. Ovviamente le Offerte associate alla stessa Asta sono riferite allo stesso Articolo; quindi, la navigazione nello schema avviene solo per le Offerte associate ad Aste diverse.*

Concetto	Costrutto	Accessi	Tipo
Utente Base	E	1	L
U-O	R	8	L
Offerta	E	8	L
O-A	R	8	L
Asta	E	8	L
Collegamento	R	8	L
Articolo	E	8	L

Costo singola operazione:  $(1 + 8*6) = 49$  accessi

Accessi totali:  $49 * 1500 = 73.500$  accessi/giorno

## Ristrutturazione dello schema E-R

Una volta terminata l’analisi dei costi delle singole operazioni è possibile iniziare la ristrutturazione dello schema E-R per poter procedere alla traduzione verso lo schema logico. In particolare, sono presenti varie ridondanze nello schema che potrebbero essere rimosse o mantenute. La prima ridondanza riguarda l’attributo *Intestatario* sull’entità *Carta di Credito*, che si decide di rimuovere dallo schema ristrutturato. Questa decisione è derivata dal fatto che l’associazione che mette in relazione ogni *Utente Base* con la propria *Carta di Credito* ha cardinalità uno-a-uno. Nella traduzione verso lo schema logico ogni relazione di *Utente Base* conterrà anche la chiave che identifica la *Carta di Credito* (numero di carta), rendendo inutile mantenere questa informazione.

Poi sono stati individuati altri due dati ridondanti nello schema E-R che sono collegati all'entità *Asta*: gli attributi *OffertaMassima* e *NumOfferte* (numero totale offerte). Questi due attributi sono ridondanti perché è possibile calcolarli a partire dalle occorrenze di *Offerte* per una singola *Asta*. L'analisi dei costi delle singole operazioni effettuata precedentemente ci permette di stimare i costi prima e dopo la rimozione delle seguenti ridondanze. In particolare, si notano due operazioni ad essere dipendenti da questi due attributi ridondanti, tra quelle evidenziate dalla *tavola delle operazioni*: quella di *visualizzazione dello stato di un'asta* (codice operazione UA4) e quella di *registrazione di una nuova offerta su un'asta attiva* (codice operazione UB6).

Analizziamo i costi prima e dopo la rimozione delle ridondanze in entrambi i casi. Iniziamo dall'operazione UA4:

*Operazione UA4 – Visualizza lo stato di un'asta*

*Si intende la visualizzazione di tutti i dettagli di una singola asta, inclusi il totale delle offerte e il valore della massima offerta. Quando si visualizzano i dettagli di un'asta attiva, vengono mostrati anche l'articolo associato e la categoria.*

Concetto	Costrutto	Accessi	Tipo
Asta	E	1	L
Articolo	E	1	L
Collegamento	R	1	L
ART-C	R	1	L
Categoria	E	1	L

Costo singola operazione:  $5 \cdot 1 = 5$  accessi

Accessi totali:  $5 \cdot 500 = 2500$  accessi/giorno

Adesso procediamo all'analisi dei costi andando a rimuovere entrambi gli attributi ridondanti:

*Operazione UA4 (ristrutturata) – Visualizza lo stato di un'asta*

*Si suppone che ad ogni asta abbiano partecipato mediamente 10 utenti durante tutta la durata e che ognuno di essi abbia fatto circa 3 offerte, tranne 2 che ne hanno fatte 5 (per rispettare l'assunzione iniziale che il 20% degli Utenti Base siano particolarmente attivi).*

Concetto	Costrutto	Accessi	Tipo
Asta	E	1	L

Articolo	E	1	L
Collegamento	R	1	L
ART-C	R	1	L
Categoria	E	1	L
O-A	R	34	L
Offerta	E	34	L

Costo singola operazione:  $5 + 34 + 34 = 73$  accessi

Accessi totali:  $73 * 500 = 36.500$  accessi/giorno

Ora analizziamo i costi prima e dopo la rimozione delle ridondanze per l'operazione UB5:

*Operazione UB5 – Registra offerta su articolo in asta*

*Si suppone la presenza degli attributi ridondanti Offerta massima e Numero Offerte sull'entità Asta. Questi attributi dovranno essere aggiornati ogni volta che un Utente Base aggiunge una nuova Offerta ad un'Asta; in particolare, il valore di Numero Offerte viene incrementato ad ogni nuova Offerte di una unità e il valore di Offerta massima viene aggiornato con l'ultima Offerta aggiunta per l'Asta che avrà importo sicuramente maggiore di tutte le precedenti.*

Concetto	Costrutto	Accessi	Tipo
Offerta	E	1	S
O-A	R	1	S
U-O	R	1	S
Asta	E	1	L
Asta	E	2	S

Costo singola operazione:  $(1 + 1 + 1 + 2) * 2 + 1 * 1 = 11$  accessi

Accessi totali:  $11 * 150 = 1650$  accessi/ora = 39.600 accessi/giorno

*Operazione UB5 (ristrutturata) – Registra offerta su articolo in asta*

Concetto	Costrutto	Accessi	Tipo
Offerta	E	1	S
O-A	R	1	S
U-O	R	1	S

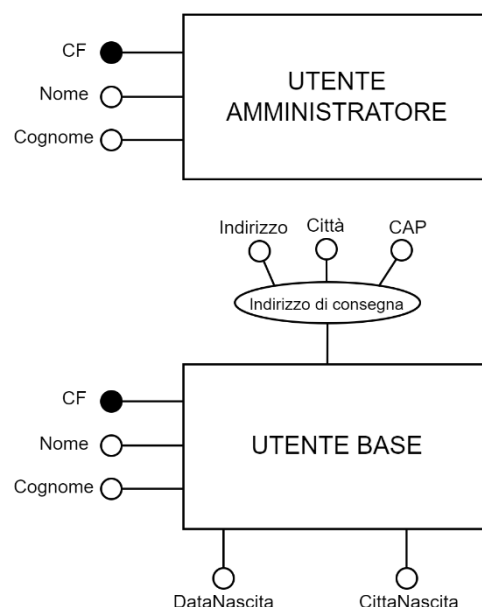
Costo singola operazione:  $(1 + 1 + 1) * 2 = 6$  accessi

Accessi totali:  $6 * 150 = 900$  accessi/ora = 21.600 accessi/giorno

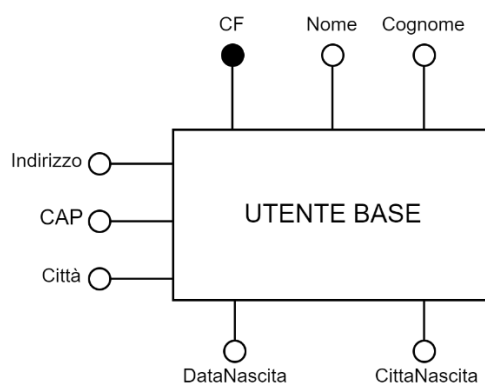
I costi totali con le ridondanze risultano: 42.100 accessi/giorno. Mentre con le ridondanze rimosse si hanno: 58.100 accessi/giorno.

Dall'analisi effettuata, si osserva che l'eliminazione dei due dati ridondanti comporterebbe un significativo aumento del numero totale di accessi giornalieri. Pertanto, si decide di mantenere entrambe le ridondanze per garantire un migliore bilanciamento tra efficienza e accesso ai dati.

Proseguendo con la ristrutturazione, è stata eliminata la generalizzazione di *Utente* in *Utente Base* e *Utente Amministratore* mediante accorpamento del padre nei figli. Questa scelta è motivata dalla netta separazione funzionale tra le due tipologie di utenti, che presentano ruoli, privilegi e relazioni completamente distinti all'interno del sistema. Di conseguenza, l'entità padre *Utente* è stata rimossa e i suoi attributi comuni sono stati replicati nelle entità figlie.



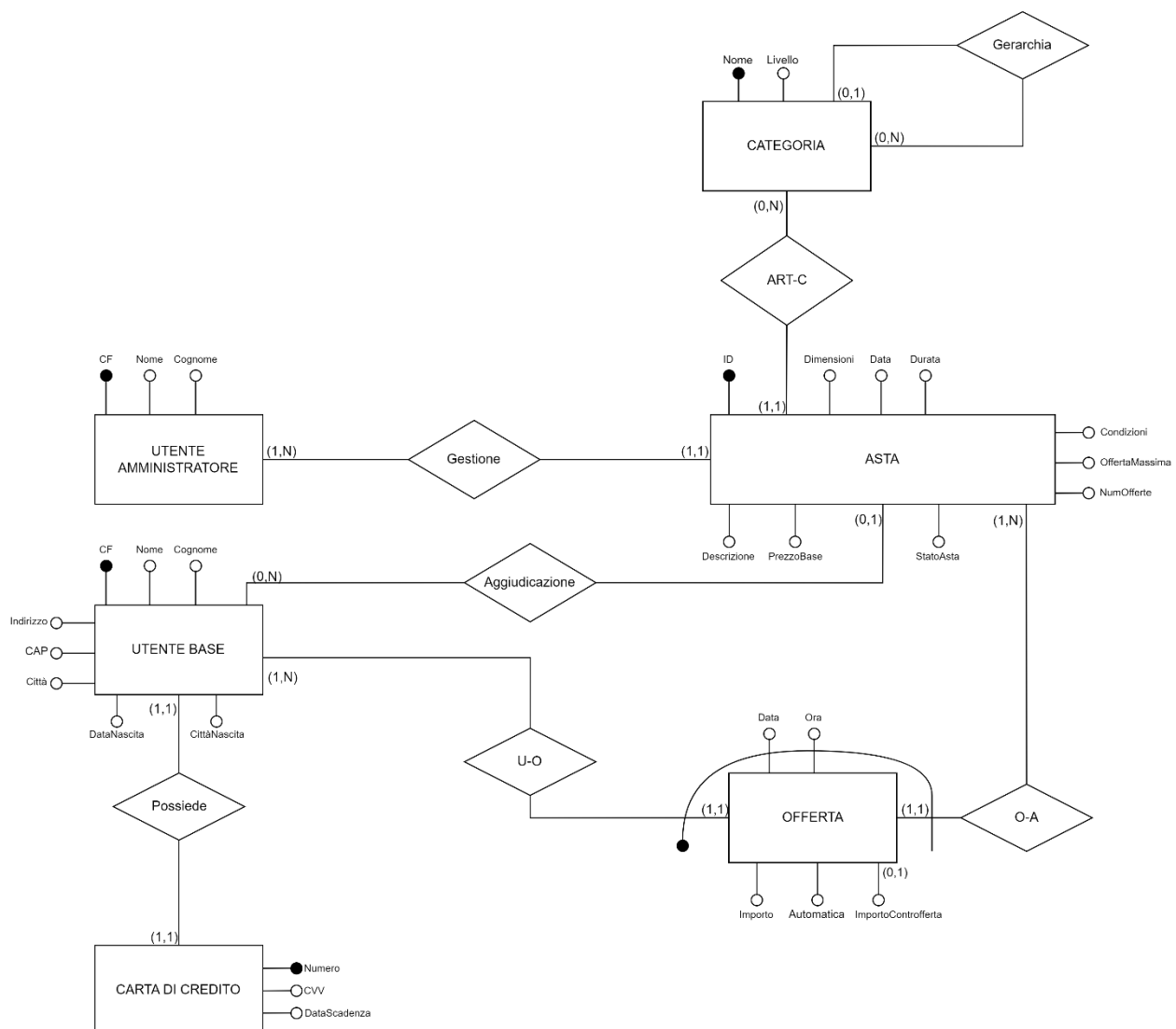
Inoltre, è stato rimosso l'attributo composto *Indirizzo di Consegna* dell'entità *Utente Base* ed è stato separato in tre diversi attributi: *Indirizzo*, *CAP* e *Città*.



Nello schema concettuale non sono presenti attributi multivalore da eliminare o partizionamenti di entità da eseguire. Anche la scelta degli identificatori principali è stata già eseguita correttamente durante la progettazione concettuale.

L'ultima fase della ristrutturazione riguarda l'analisi dell'accorpamento tra *Articolo* e *Asta*. Queste entità presentano una forte dipendenza funzionale: ogni asta è univocamente associata a un articolo e viene inizializzata contestualmente alla sua pubblicazione. L'accorpamento in un'unica entità *Asta* eliminerebbe la necessità di navigare l'associazione *Collegamento*, riducendo significativamente gli accessi necessari per le operazioni più frequenti del sistema.

Lo schema concettuale ristrutturato con l'accorpamento di *Asta* e *Articolo* è il seguente:



Procediamo ora con l'analisi dei costi delle operazioni prima e dopo l'accorpamento, in modo da poter decidere se finalizzare tale schema ristrutturato.

Le operazioni UA1 (Inserisci nuovo articolo) e UA3 (Inizializza asta per articolo) vengono accorpate in un'unica operazione UA1 (Pubblica nuova asta).

Si riportano qui di seguito le operazioni che vengono influenzate da tale accorpamento: UA1, UA3, UB4, UB1, UB2, UB3, UB4 e UB6.

Gli accessi prima di accorpare le due entità risultano pari a 906.795 accessi/ora.

Adesso si mostrano le nuove analisi dei costi delle operazioni dopo aver accorpato le due entità:

*Operazione UA1 (dopo accorpamento entità) – Pubblica nuova asta*

*Si considera la pubblicazione quotidiana di circa 10 nuovi articoli da parte di ogni Utente Amministratore.*

Concetto	Costrutto	Accessi	Tipo
Articolo	E	1	S
Gestione	R	1	S

Costo singola operazione:  $2 \times 2 = 4$  accessi

Accessi totali:  $4 \times 300 = 1200$  accessi/giorno

*Operazione UA4 (dopo accorpamento entità) – Visualizza lo stato di un'asta*

Concetto	Costrutto	Accessi	Tipo
Asta	E	1	L
ART-C	R	1	L
Categoria	E	1	L

Costo singola operazione:  $3 \times 1 = 3$  accessi

Accessi totali:  $3 \times 500 = 1500$  accessi/giorno

*Operazione UB1 (dopo accorpamento entità) – Visualizza le aste attive*

*Si considera l'inserimento di circa 300 aste al giorno. Inoltre, si considera una durata media di 4 giorni per ogni asta pubblicata, avendo quindi una media di 1200 aste attive al giorno.*

Concetto	Costrutto	Accessi	Tipo
Asta	E	1200	L

Costo singola operazione:  $1200 \times 1 = 1200$  accessi

Accessi totali:  $1200 \times 250 = 300.000$  accessi/ora



*Operazione UB2 (**dopo accorpamento entità**)* – Visualizza le aste attive per categoria

*Si assume la ricerca basata su Categorie foglia (circa 168) e che le aste attive quotidiane siano distribuite equamente tra queste categorie (circa 7 aste per categoria).*

Concetto	Costrutto	Accessi	Tipo
Categoria	E	1	L
ART-C	R	7	L
Asta	E	7	L

Costo singola operazione:  $(7 + 7 + 1) * 1 = 15$  accessi

Accessi totali:  $15 * 100 = 1500$  accessi/ora

*Operazione UB3 (**dopo accorpamento entità**)* – Visualizza le aste attive per utente amministratore

*Si assume che ogni Utente Amministratore pubblichi quotidianamente 10 aste. Si assume anche che le aste abbiano durata media di circa 4 giorni.*

Concetto	Costrutto	Accessi	Tipo
Utente Amministratore	E	1	L
Gestione	R	40	L
Asta	E	40	L

Costo singola operazione:  $(40 * 2 + 1) * 1 = 81$  accessi

Accessi totali:  $81 * 80 = 6480$  accessi/giorno

*Operazione UB4 (**dopo accorpamento entità**)* – Visualizza tutti gli articoli aggiudicati da un Utente Base

*Si assume che un singolo Utente Base si aggiudichi circa 10 aste al mese. Essendo questa un'operazione che viene fatta quotidianamente si assume anche che al momento dell'esecuzione un singolo Utente Base si sia aggiudicato circa la metà delle aste mediamente aggiudicate in un intero mese.*

**NOTA:** viene mostrato solo l'aggiornamento dei nomi nella tabella. In questo caso non c'è nessuna variazione sui costi.

Concetto	Costrutto	Accessi	Tipo
Utente Base	E	1	L
Aggiudicazione	R	5	L
Asta	E	5	L

Costo singola operazione:  $11 * 1 = 11$  accessi

Accessi totali:  $11 * 200 = 2200$  accessi/giorno

*Operazione UB6 (**dopo accorpamento entità**)* – Visualizza elenco articoli associati ad un’asta attiva sui quali hai effettuato un’offerta

*Si assume che ogni Utente Base partecipi attivamente a 2 aste al giorno; dunque, è necessario considerare anche le aste dei giorni precedenti che sono ancora in corso. Assumendo che un'asta abbia una durata media di 4 giorni, l'Utente Base potrebbe avere Offerte attive su Aste iniziate fino a 3 giorni prima. Ovviamente le Offerte associate alla stessa Asta sono riferite allo stesso Articolo; quindi, la navigazione nello schema avviene solo per le Offerte associate ad Aste diverse.*

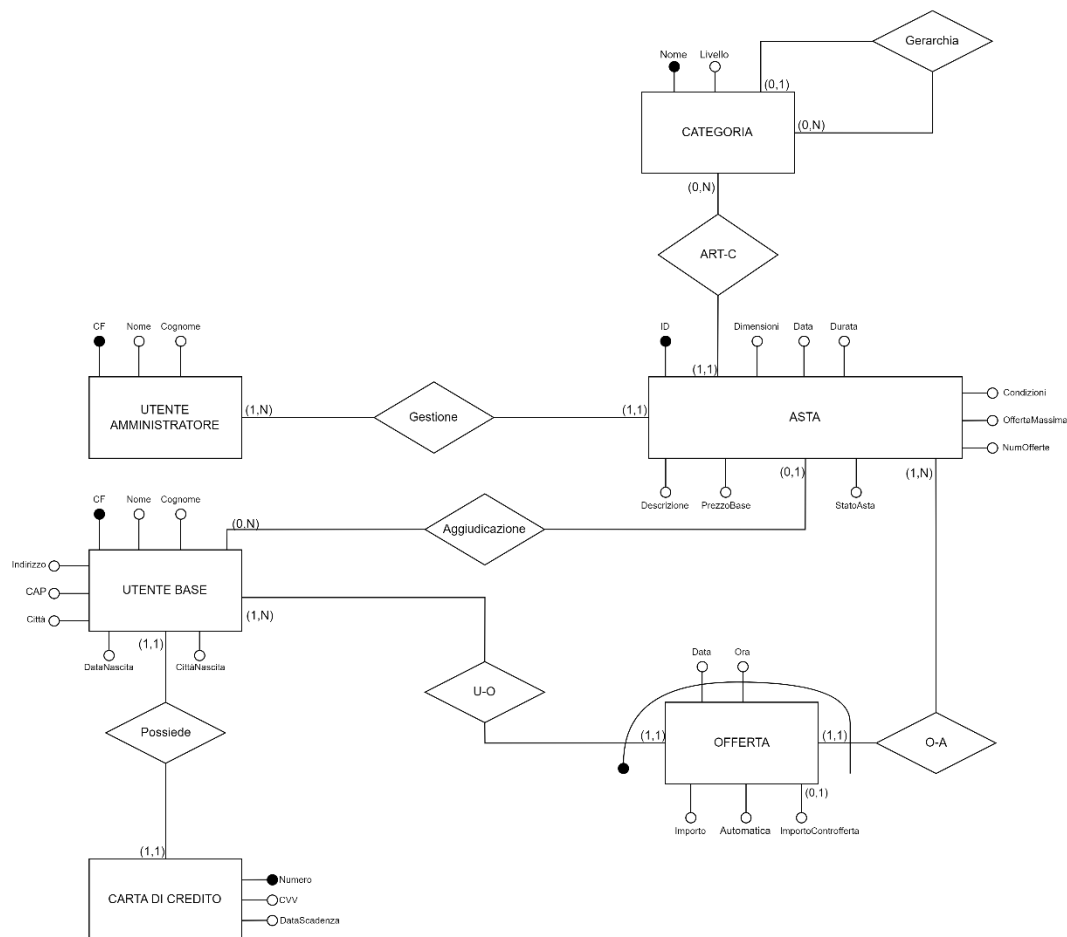
Concetto	Costrutto	Accessi	Tipo
Utente Base	E	1	L
U-O	R	8	L
Offerta	E	8	L
O-A	R	8	L
Asta	E	8	L

Costo singola operazione:  $8 * 4 + 1 = 33$  accessi

Accessi totali:  $33 * 1500 = 49.500$  accessi/giorno

Dopo aver effettuato l’accorpamento tra le entità Asta e Articolo, la somma degli accessi risulta pari a *304.036 accessi/ora*. Dunque, l’accorpamento ha ridotto gli accessi orari di circa il 67% sulle sole operazioni relative alle entità *Asta* e *Articolo*.

Lo schema finale ristrutturato risulta quindi:



## Trasformazione di attributi e identificatori

Nello schema concettuale non sono presenti attributi ripetuti o identificatori esterni da modificare.

## Traduzione di entità e associazioni

UTENTE~~BASE~~ (CF, Nome, Cognome, Indirizzo, CAP, Città, DataNascita, CittàNascita, CartaCredito)

~~GESTIONE~~ (ID, ~~UtenteAmministratore~~)

CARTADICREDITO (Numero, CVV, DataScadenza)

~~POSSIEDE~~ (~~UtenteBase~~, ~~CartaCredito~~)

UTENTEAMMINISTRATORE (CF, Nome, Cognome)

OFFERTA (UtenteBase, Asta, Data, Ora, Importo, Automatica, ImportoControfferta\*)

ASTA (ID, Dimensioni, Data, Durata, Descrizione, PrezzoBase, StatoAsta, NumOfferte, OffertaMassima, CondizioniArticolo, Categoria, UtenteAmministratore, UtenteBase\*)

CATEGORIA (Nome, Livello, CategoriaSuperiore\*)

**AGGIUDICAZIONE** (Asta, UtenteBase)

**ART-C** (Asta, Categoria)

La traduzione finale delle entità e delle associazione risulta essere:

**UTENTEBase** (CF, Nome, Cognome, Indirizzo, CAP, Città, DataNascita, CittàNascita, CartaCredito)

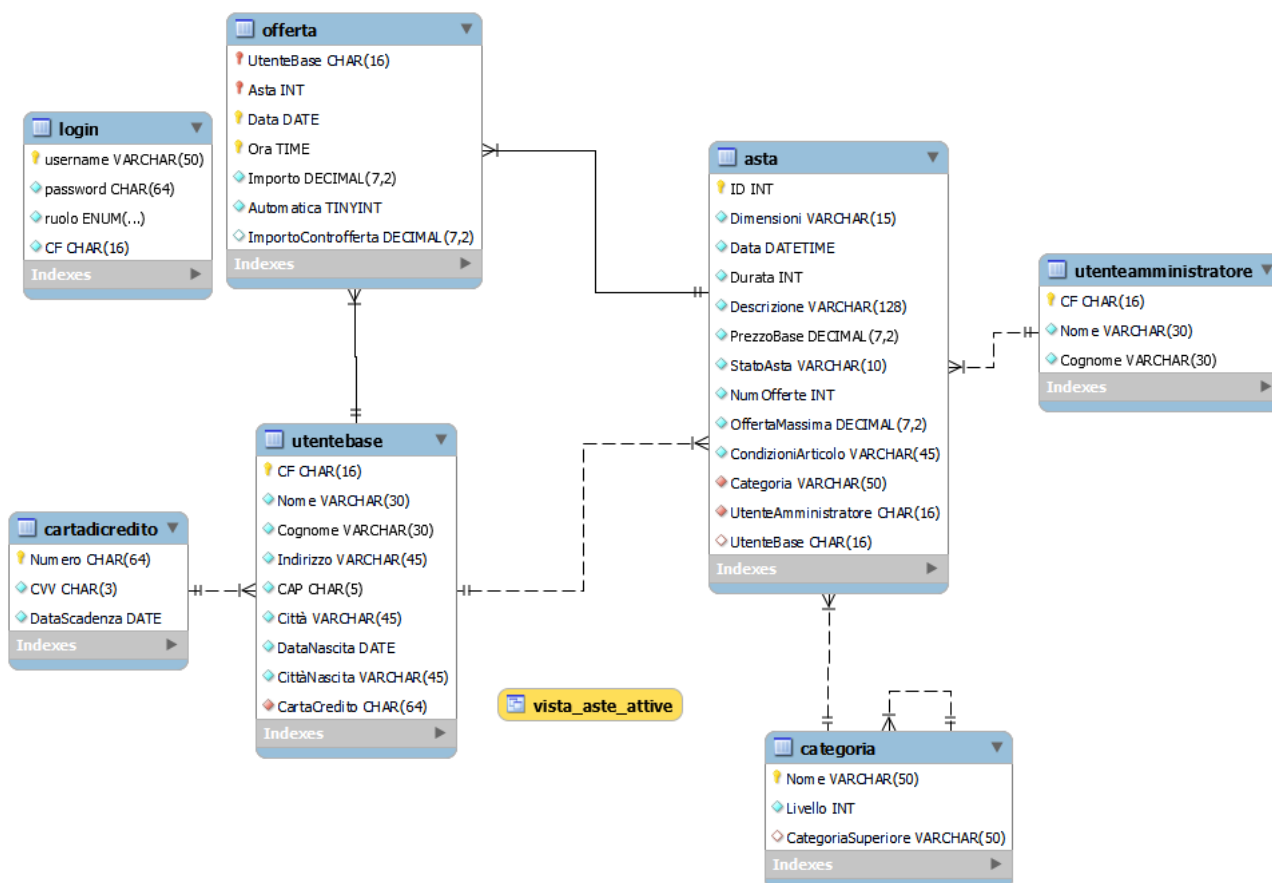
**CARTADICREDITO** (Numero, CVV, DataScadenza)

**UTENTEAMMINISTRATORE** (CF, Nome, Cognome)

**OFFERTA** (UtenteBase, Asta, Data, Ora, Importo, Automatica, ImportoControfferta\*)

**ASTA** (ID, Dimensioni, Data, Durata, Descrizione, PrezzoBase, StatoAsta, NumOfferte, OffertaMassima, CondizioniArticolo, Categoria, UtenteAmministratore, UtenteBase\*)

**CATEGORIA** (Nome, Livello, CategoriaSuperiore\*)



## Normalizzazione del modello relazionale

Per verificare se il modello relazionale che si è ottenuto risulta qualitativamente buono è necessario verificare la soddisfazione delle forme normali, in particolare della forma normale di Boyce-Codd (BCNF).

Per ogni relazione dello schema logico, è stata preliminarmente verificata la presenza delle dipendenze funzionali tra gli attributi e la conformità alle forme normali.

Tutte le relazioni risultano in BCNF poiché, per ogni dipendenza funzionale  $X \rightarrow Y$ ,  $X$  contiene una chiave della relazione. In particolare:

- le chiavi primarie determinano univocamente tutti gli altri attributi;
- non esistono dipendenze funzionali tra attributi non chiave;
- non sono presenti dipendenze transitive.

Gli attributi *OffertaMassima* e *NumOfferte* nella relazione *Asta*, pur essendo derivabili dalle occorrenze di *Offerta*, sono stati mantenuti come ridondanze controllate. Questa scelta:

- non viola la BCNF, in quanto dipendono funzionalmente solo dalla chiave primaria ID di *Asta*;
- è giustificata dall'analisi dei costi che dimostra un miglioramento delle prestazioni del 38%.

Il modello relazionale risulta completamente normalizzato secondo la BCNF pur mantenendo ridondanze controllate per ottimizzare le prestazioni generali del sistema.

## 5. Progettazione fisica

### Utenti e privilegi

Nel sistema sono previsti 3 *tipi di utenti*:

1. **Login**: identificato con la sigla *aste\_login*, si occupa esclusivamente di gestire la procedura di login e di registrazione, quindi è deputato principalmente all'autenticazione.
2. **Utente Amministratore**: identificato con la sigla *aste\_admin*, si occupa di gestire le aste e di gestire le categorie da assegnare ad esse.
3. **Utente Base**: identificato con la sigla *aste\_user*, può fare offerte su aste attive, visualizzare le aste attive, visualizzare gli articoli che si è aggiudicato in passato vincendo le aste e visualizzare gli articoli per i quali è presente un'asta attiva a cui ha fatto almeno un'offerta.

La scelta di aggiungere un utente di *Login* è derivata dalla necessità di implementare il *Principle of Least Privilege*, per garantire una maggiore sicurezza e stabilità del sistema, grazie alla separazione dei privilegi e la minimizzazione del rischio di accessi non autorizzati ai dati sensibili del sistema. Inoltre, viene applicato anche il principio *Need-to-Know*, perché in questo modo ogni utente ha un proprio ruolo definito e ha la possibilità di accedere solamente alle informazioni richieste per lo svolgimento delle proprie funzioni.

Per memorizzare le credenziali di accesso è stato necessario aggiungere una tabella di *Login*, in modo da mantenere un riferimento tra l'utente del sistema e le proprie credenziali di accesso. Inoltre, le password sono memorizzate in forma cifrata (sha256) per garantire riservatezza e sicurezza dei dati sensibili.

Sono stati creati i tre nuovi utenti e ad essi sono state assegnate le varie procedure che possono andare ad eseguire:

#### 1. Utente Login (*aste\_login*)

Ha accesso limitato alle sole procedure necessarie per:

- Autenticazione utenti (*login*)
- Registrazione nuovi utenti (*registerUser*)
- Recupero dettagli utente per autenticazione (*dettagli\_utente\_cf*)

#### 2. Utente Amministratore (*aste\_admin*)

Le procedure assegnate corrispondono alle operazioni della tavola delle operazioni:

- **UA1-UA3:** Pubblica nuova asta (*creaAsta*)
- **UA2:** Inserisci nuova categoria (*aggiungiCat*)
- **UA4:** Visualizza lo stato di un'asta (*visualizza\_aste\_stato*)
- **UA5:** Visualizza le offerte associate ad un'asta (*visualizza\_offerte\_asta*)
- Procedure aggiuntive per la gestione completa: *modificaCat*, *eliminaCat*, *listaCategorie*, *chiudiAsta*.

### 3. Utente Base (aste\_user)

Le procedure assegnate implementano le operazioni:

- **UB1:** Visualizza le aste attive (*visualizzaAsteFiltrate*)
- **UB2:** Visualizza le aste attive per categoria (implementata in *visualizzaAsteFiltrate* con filtro)
- **UB3:** Visualizza le aste attive per utente amministratore (implementata in *visualizzaAsteFiltrate* con filtro)
- **UB4:** Visualizza tutti gli articoli aggiudicati (*asteVinteUtente*)
- **UB5:** Registra offerta su articolo in asta (*aggiungiOfferta*)
- **UB6:** Visualizza elenco articoli con offerte attive (*visualizzaAstePartecipate*)
- Procedure aggiuntive: *aggiorna\_indirizzo\_consegna*, *listaCategorie*, *listaAmministratori*, *getMigliorOfferta*, *getControfferteAttive*.

Codice	Descrizione	Procedura	Utente
<b>L1</b>	Login	login	aste_login
<b>UA1-UA3</b>	Pubblica nuova asta	creaAsta	aste_admin
<b>UA2</b>	Inserisci nuova categoria	aggiungiCat	aste_admin
<b>UA4</b>	Visualizza lo stato di un'asta	visualizza_aste_stato	aste_admin
<b>UA5</b>	Visualizza le offerte associate	visualizza_offerte_asta	aste_admin
<b>UB1, UB2, UB3</b>	Visualizza aste (con filtri)	visualizzaAsteFiltrate	aste_user
<b>UB4</b>	Visualizza articoli aggiudicati	asteVinteUtente	aste_user
<b>UB5</b>	Registra offerta	aggiungiOfferta	aste_user
<b>UB6</b>	Visualizza aste partecipate	visualizzaAstePartecipate	aste_user

## Strutture di memorizzazione

<b>Tabella Login</b>		
<b>Colonna</b>	<b>Tipo di dato</b>	<b>Attributi<sup>4</sup></b>
<b>Username</b>	VARCHAR (50)	PK, NN
<b>Password</b>	CHAR(64)	NN
<b>Ruolo</b>	ENUM ('ADMIN', 'USER')	NN
<b>CF</b>	CHAR(16)	NN

<b>Tabella UtenteBase</b>		
<b>Colonna</b>	<b>Tipo di dato</b>	<b>Attributi</b>
<b>CF</b>	CHAR (16)	PK, NN
<b>Nome</b>	VARCHAR (30)	NN
<b>Cognome</b>	VARCHAR (30)	NN
<b>Indirizzo</b>	VARCHAR (45)	NN
<b>CAP</b>	CHAR (5)	NN
<b>Città</b>	VARCHAR (45)	NN
<b>DataNascita</b>	DATE	NN
<b>CittàNascita</b>	VARCHAR (45)	NN
<b>CartaCredito</b>	CHAR (64)	NN, UQ

<b>Tabella CartaDiCredito</b>		
<b>Colonna</b>	<b>Tipo di dato</b>	<b>Attributi</b>
<b>Numero</b>	CHAR (64)	PK, NN
<b>CVV</b>	CHAR (3)	NN
<b>DataScadenza</b>	DATE	NN

<b>Tabella UtenteAmministratore</b>		
<b>Colonna</b>	<b>Tipo di dato</b>	<b>Attributi</b>
<b>CF</b>	CHAR (16)	PK, NN
<b>Nome</b>	VARCHAR (30)	NN
<b>Cognome</b>	VARCHAR (30)	NN

<b>Tabella Offerta</b>		
<b>Colonna</b>	<b>Tipo di dato</b>	<b>Attributi</b>
<b>UtenteBase</b>	CHAR (16)	PK, NN
<b>Asta</b>	INT	PK, NN
<b>Data</b>	DATE	PK, NN
<b>Ora</b>	TIME	PK, NN
<b>Importo</b>	DECIMAL (7,2)	NN
<b>Automatica</b>	BOOLEAN	NN
<b>ImportoControfferta</b>	DECIMAL (7,2)	

---

<sup>4</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 Asta		
Colonna	Tipo di dato	Attributi
ID	INT	PK, NN, AI
Dimensioni	VARCHAR (15)	NN
Data	DATETIME	NN
Durata	INT	NN
Descrizione	VARCHAR (128)	NN
PrezzoBase	DECIMAL (7,2)	NN
StatoAsta	VARCHAR (10)	NN
NumOfferte	INT	NN
OffertaMassima	DECIMAL (7,2)	NN
CondizioniArticolo	VARCHAR (45)	NN
Categoria	VARCHAR (50)	NN
UtenteAmministratore	CHAR (16)	NN
UtenteBase	CHAR (16)	

Tabella Categoria		
Colonna	Tipo di dato	Attributi
Nome	VARCHAR (50)	PK, NN
Livello	INT	NN
CategoriaSuperiore	VARCHAR (50)	

## Indici

Gli indici esposti qui di seguito sono stati aggiunti per motivi prestazionali, facendo riferimento all'analisi dei costi delle singole operazioni, svolta nella fase di progettazione logica.

In particolare, i principali indici sono stati creati per migliorare le prestazioni delle operazioni più frequenti e costose del sistema.

Tabella Asta	
Indice idx_asta_stato	Tipo <sup>5</sup> :
StatoAsta	IDX

Questo indice è stato aggiunto per ottimizzare la *vista\_aste\_attive* per assolvere l'operazione UB1 (*Visualizza aste attive*) che genera, dopo l'accorpamento, 300.000 accessi/ora.

Tabella Asta	
Indice idx_asta_amministratore_stato	Tipo <sup>6</sup> :

<sup>5</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

UtenteAmministratore, StatoAsta	IDX
---------------------------------	-----

Questo indice è stato creato per ottimizzare l'operazione UA4 (*Visualizza lo stato di un'asta*) che ha 500/accessi giornalieri. L'indice ha lo scopo di filtrare in modo efficiente le aste per amministratore e poi per stato.

Tabella Asta	
Indice idx_asta_categoria	Tipo <sup>7</sup> :
Categoria	IDX

Questo indice serve per ottimizzare l'operazione UB2 (*Visualizza aste attive per categoria*) con 1500 accessi/ora.

Tabella Asta	
Indice idx_utente_stato	Tipo <sup>8</sup> :
UtenteBase, StatoAsta	IDX

L'indice *idx\_utente\_stato* migliora le prestazioni dell'operazione UB4 (*Visualizza articoli aggiudicati*) che genera 2200 accessi/giorno, andando a trovare velocemente le aste terminate vinte da un certo utente (alta cardinalità).

Tabella Offerta	
Indice idx_offerta_asta_importo	Tipo <sup>9</sup> :
Asta, Importo, Data, Ora	IDX

Questo indice serve per ottimizzare le operazioni UA5 (*Visualizza offerte asta*) e l'operazione UB5 (*Registra offerta*). In questo modo viene migliorata la ricerca delle offerte per una data asta (alta cardinalità) e viene ottimizzato l'ordinamento per data e ora.

Tabella Offerta	
Indice idx_offerta_utente_asta	Tipo <sup>10</sup> :

---

<sup>6</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>7</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>8</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>9</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>10</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

UtenteBase, Asta	IDX
------------------	-----

L'indice supporta l'operazione UB6 (*Visualizza aste partecipate*), permettendo di trovare velocemente tutte le offerte di un utente raggruppate per asta.

Tabella Offerta	
Indice idx_offerta_controfferta	Tipo <sup>11</sup> :
Asta, ImportoControfferta	IDX

Questo indice ottimizza la procedura *getControfferteAttive* per il sistema di gestione dei rilanci automatici, permettendo di andare ad identificare rapidamente le controfferte superiori all'importo massimo corrente.

Infine, è stato aggiunto anche un indice di tipo *unique* per l'attributo *CartaCredito* della relazione *UtenteBase*, poiché ogni carta di credito deve essere associata ad un singolo utente e, di conseguenza, il suo numero deve risultare univoco.

Tabella UtenteBase	
Indice carta_credito_unique	Tipo <sup>12</sup> :
CartaCredito	UQ

## Trigger

I trigger sono stati utilizzati per controllare che vengano rispettate le condizioni imposte dalle regole aziendali, sia di vincolo che di derivazione, in modo da garantire il corretto funzionamento del sistema.

Di seguito vengono elencati tutti i trigger aggiunti nello schema:

- vincolo sulla *Durata* imposto alla creazione di una nuova *Asta*

```
CREATE TRIGGER controllo_durata_asta
BEFORE INSERT ON asta
```

---

<sup>11</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>12</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

```
FOR EACH ROW
BEGIN
    IF NEW.Durata < 1 OR NEW.Durata > 7 THEN
        SIGNAL SQLSTATE '45101'
        SET MESSAGE_TEXT = 'La durata dell\'asta deve essere compresa tra 1 e 7 giorni.';
    END IF;
END
```

- vincolo sul valore minimo del *Prezzo Base* alla creazione di una nuova *Asta*

```
CREATE TRIGGER controllo_prezzo_base
BEFORE INSERT ON asta
FOR EACH ROW
BEGIN
    IF NEW.PrezzoBase <= 0 THEN
        SIGNAL SQLSTATE '45102'
        SET MESSAGE_TEXT = 'Il prezzo base deve essere maggiore di 0.';
    END IF;
END
```

- impedisce inserimento di una categoria di livello superiore al livello massimo

```
CREATE TRIGGER controllolivello_categoria
BEFORE INSERT ON categoria
FOR EACH ROW
BEGIN
    IF NEW.Livello > 3 THEN
        SIGNAL SQLSTATE '45300'
        SET MESSAGE_TEXT = 'Non è possibile inserire una categoria con livello superiore a 3.';
    END IF;
END
```

- imporre che l'importo della prima *Offerta* sia maggiore del valore dell'attributo *PrezzoBase* dell'*Asta*

```
CREATE TRIGGER verifica_importo_offerta
BEFORE INSERT ON offerta
FOR EACH ROW
BEGIN
    DECLARE prezzo_base DECIMAL(7,2);
    DECLARE num_offerte INT;
    SELECT PrezzoBase INTO prezzo_base FROM asta WHERE ID = New.Asta;
    SELECT NumOfferte INTO num_offerte FROM asta WHERE ID = New.Asta;

    IF num_offerte = 0 AND New.Importo <= prezzo_base THEN
        SIGNAL SQLSTATE '45003'
        SET MESSAGE_TEXT = 'L\'offerta deve essere maggiore del prezzo base.';
    END IF;
END
```

- imporre che l'importo di ogni nuova offerta sia sempre maggiore dell'offerta massima attuale + 0,50€

```
CREATE TRIGGER verifica_incremento_offerta
BEFORE INSERT ON offerta
FOR EACH ROW
BEGIN
    DECLARE offerta_massima_corrente DECIMAL(7,2) DEFAULT 0.00;
    DECLARE incremento_minimo DECIMAL(7,2) DEFAULT 0.50;

    -- recupera l'offerta massima attuale per l'asta
    SELECT OffertaMassima
    INTO offerta_massima_corrente
    FROM asta
    WHERE ID = NEW.Asta;
```

```

    IF offerta_massima_corrente > 0 AND NEW.Importo < (offerta_massima_corrente +
    incremento_minimo) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'L\'offerta deve essere almeno 0,50€ superiore all\'offerta
    massima attuale.';
    END IF;
END

```

- aggiornamento automatico *OffertaMassima* e *UtenteBase* dopo ogni inserimento nuova *Offerta*

```

CREATE TRIGGER aggiornamento_offerta_vincitore
AFTER INSERT ON offerta
FOR EACH ROW
BEGIN
    UPDATE Asta
    SET
        OffertaMassima = New.Importo,
        UtenteBase = New.UtenteBase,
        NumOfferte = NumOfferte + 1
    WHERE ID = New.Asta;
END

```

- verifica stato *Asta* prima di inserimento di una nuova *Offerta*

```

CREATE TRIGGER verifica_stato_asta
BEFORE INSERT ON offerta
FOR EACH ROW
BEGIN
    DECLARE stato_asta VARCHAR(10);

    SELECT StatoAsta INTO stato_asta FROM asta WHERE ID = NEW.Asta;

```

```
IF stato_asta = 'Terminata' THEN
    SIGNAL SQLSTATE '45600'
    SET MESSAGE_TEXT = 'Non è possibile fare offerte su un\'asta terminata.';
END IF;
END
```

## Eventi

Per la rimozione automatica dei dati non più necessari nel sistema è stato creato un evento, il quale viene schedulato durante la configurazione iniziale dello schema all'inizio di ogni mese, in modo tale da ridurre il carico di lavoro. Questo evento viene eseguito automaticamente e gli utilizzatori del sistema non eseguono alcuna operazione su di esso.

Per poter eseguire correttamente l'evento, deve essere attivato lo scheduler nel DBMS:

```
SET GLOBAL event_scheduler = ON;
```

L'evento è stato chiamato "pulizia\_dati\_obsoleti" e il codice SQL per l'istanziamento è il seguente:

```
CREATE EVENT IF NOT EXISTS pulizia_dati_obsoleti
    ON SCHEDULE EVERY 1 MONTH
    STARTS '2025-01-01 00:00:00'
    COMMENT 'Eliminazione dati con più di 10 anni'
DO
BEGIN
    -- elimina offerte delle aste obsolete
    DELETE o FROM offerta o
    INNER JOIN asta a ON o.Asta = a.ID
    WHERE a.Data < DATE_SUB(CURDATE(), INTERVAL 10 YEAR);

    -- elimina aste vecchie
    DELETE FROM asta
    WHERE Data < DATE_SUB(CURDATE(), INTERVAL 10 YEAR);
END
```

Dunque, questo evento si occupa di eliminare sia le *Offerte* che le *Aste* che sono state inserite nel sistema da più di 10 anni.

Per il corretto funzionamento del sistema è previsto anche un altro evento che si occupa dell'aggiornamento dello stato di tutte le aste che terminano o che stanno iniziando.

```
CREATE EVENT IF NOT EXISTS aggiorna_stato_aste
    ON SCHEDULE EVERY 1 MINUTE
    COMMENT 'Aggiornamento automatico stato aste'
DO
BEGIN
    UPDATE asta
    SET StatoAsta = 'ATTIVA'
    WHERE StatoAsta = 'FUTURA' AND NOW() >= Data;

    UPDATE asta
    SET StatoAsta = 'TERMINATA'
    WHERE StatoAsta = 'ATTIVA' AND NOW() >= DATE_ADD(Data, INTERVAL Durata DAY);
END
```

## Viste

Le viste sono state utilizzate per permettere agli Utenti Basi di effettuare delle operazioni di visualizzazione anche non potendo direttamente accedere in lettura a determinate tabelle.

In particolare, nella specifica di progetto è specificato come un UtenteBase debba avere la possibilità di visualizzare in ogni momento tutte le aste attive e i dettagli di ognuna, escludendo però dalla visualizzazione il miglior offerente in quel dato istante. Inoltre, aver definito questa vista permetterà ad un qualunque utente di filtrare le aste attive per categoria associata o per utente amministratore che l'ha inizializzata.

```
CREATE OR REPLACE VIEW vista_aste_attive AS
SELECT
    ID,
    Descrizione,
```



```
Dimensioni,  
Data,  
Durata,  
Categoria,  
PrezzoBase,  
StatoAsta,  
NumOfferte,  
OffertaMassima,  
CondizioniArticolo,  
UtenteAmministratore,  
TIMESTAMPDIFF(SECOND, NOW(), DATE_ADD(Data, INTERVAL Durata DAY)) AS  
TempoRimanenteSec  
FROM asta  
WHERE StatoAsta = 'ATTIVA'  
ORDER BY TempoRimanenteSec;
```

In questo modo vengono mostrate tutte le tuple della tabella *Aste* che fanno riferimento ad occorrenze di aste attive, escludendo l'attributo *UtenteBase* per nascondere l'attuale miglior offerente. In particolare, viene calcolato in tempo reale il tempo rimanente prima della chiusura effettiva dell'asta (espresso in secondi), tradotto in formato hh:mm:ss lato client. Inoltre, il risultato viene visualizzato ordinato in modo crescente in base ai secondi rimanenti alla chiusura.

## Stored Procedures e transazioni

Le stored procedure implementate seguono le operazioni definite nella tavola delle operazioni, organizzate per codice operazione e ruolo utente.

### Operazioni di Login (L)

#### L1 - login

Implementa l'autenticazione degli utenti verificando username e password (con hash SHA2-256) nella tabella login. Non richiede transazione esplicita trattandosi di una semplice lettura. Restituisce il ruolo e il codice fiscale dell'utente autenticato.

PROCEDURA login:

```
CREATE PROCEDURE `login`(  
    in var_username VARCHAR(50),  
    in var_password VARCHAR(32),  
    out var_ruolo ENUM('ADMIN', 'USER'),  
    out var_CF CHAR(16)  
)  
BEGIN  
    DECLARE var_check_username VARCHAR(50);  
  
    SELECT username, ruolo, CF INTO var_check_username, var_ruolo, var_CF  
    FROM login  
    WHERE username = var_username AND password = sha2(var_password, 256);  
  
    IF var_check_username IS NULL THEN  
        SIGNAL SQLSTATE '45000' SET message_text = 'Credenziali di accesso errate';  
    END IF;  
END
```

### **Operazioni Utente Amministratore (UA)**

**UA1+UA3 - creaAsta** (Operazioni accorpate: Inserisci nuovo articolo + Inizializza asta)

A seguito dell'accorpamento delle entità Articolo e Asta, questa procedura unifica le operazioni UA1 e UA3. Utilizza una transazione con isolation level READ COMMITTED per garantire atomicità dell'operazione mantenendo performance ottimali. L'handler di errore cattura e propaga i messaggi SQL per il debug.

#### **PROCEDURA creaAsta:**

```
CREATE PROCEDURE `creaAsta`(  
    IN var_Dimensioni VARCHAR(15),  
    IN var_Data DATETIME,  
    IN var_Durata INT,  
    IN var_Descrizione VARCHAR(128),
```

```
IN var_PrezzoBase DECIMAL(7,2),
IN var_StatoAsta VARCHAR(10),
IN var_CondizioniArticolo VARCHAR(45),
IN var_Categoria VARCHAR(50),
IN var_UtenteAmministratore CHAR(16)
)
BEGIN
  DECLARE errno INT DEFAULT 0;
  DECLARE msg TEXT;

  DECLARE EXIT HANDLER FOR SQLEXCEPTION
  BEGIN
    GET DIAGNOSTICS CONDITION 1
      errno = MYSQL_ERRNO,
      msg = MESSAGE_TEXT;
    ROLLBACK;

    SIGNAL SQLSTATE '45201' SET MESSAGE_TEXT = msg;
  END;

  SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
  START TRANSACTION;

  INSERT INTO asta (
    Dimensioni, Data, Durata, Descrizione,
    PrezzoBase, StatoAsta,
    CondizioniArticolo, Categoria,
    UtenteAmministratore
  )
  VALUES (
    var_Dimensioni, var_Data, var_Durata, var_Descrizione,
    var_PrezzoBase, var_StatoAsta,
    var_CondizioniArticolo, var_Categoria,
    var_UtenteAmministratore
```

```
);
```

```
COMMIT;
```

```
END
```

### **UA2 - Gestione categorie**

Le operazioni sulle categorie richiedono particolare attenzione per la gestione della gerarchia:

#### **PROCEDURA aggiungiCat:**

```
CREATE PROCEDURE `aggiungiCat`(  
    in var_nome VARCHAR(50),  
    in var_livello INT,  
    in var_catsup VARCHAR(50)  
)  
BEGIN  
    SET TRANSACTION ISOLATION LEVEL READ COMMITTED;  
    START TRANSACTION;  
  
    INSERT INTO categoria (  
        Nome, Livello, CategoriaSuperiore  
    )  
    VALUES (  
        var_nome, var_livello, var_catsup  
    );  
  
    COMMIT;  
END
```

Utilizza READ COMMITTED come isolation level per ottimizzare gli inserimenti e non bloccare i dati in lettura dato che avviene una semplice insert nel database.

#### **PROCEDURA modificaCat:**

```
CREATE PROCEDURE `modificaCat`(  
    IN old_nome VARCHAR(50),  
    IN new_nome VARCHAR(50)  
)  
BEGIN  
    SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;  
    START TRANSACTION;  
  
    -- inserisco una categoria temporanea  
    INSERT INTO categoria (Nome, Livello, CategoriaSuperiore) VALUES ('TEMP_CAT', 1,  
NULL);  
  
    -- aggiorno i figli ad un valore temporaneo  
    UPDATE categoria  
        SET CategoriaSuperiore = 'TEMP_CAT'  
        WHERE CategoriaSuperiore = old_nome;  
  
    -- aggiorno il padre  
    UPDATE categoria  
        SET Nome = new_nome  
        WHERE Nome = old_nome;  
  
    -- aggiorno i figli al valore nuovo del padre  
    UPDATE categoria  
        SET CategoriaSuperiore = new_nome  
        WHERE CategoriaSuperiore = 'TEMP_CAT';  
  
    -- elimino la categoria temporanea  
    DELETE FROM categoria WHERE Nome = 'TEMP_CAT';  
  
    COMMIT;  
END
```

Procedura particolarmente complessa che utilizza una categoria temporanea per evitare violazioni di foreign key durante l'aggiornamento. Il livello `SERIALIZABLE` garantisce che nessun'altra transazione possa interferire durante la modifica dei riferimenti padre-figlio.

PROCEDURA eliminaCat:

```
CREATE PROCEDURE `eliminaCat`(  
    IN nome_categoria VARCHAR(50),  
    OUT num_del INT,  
    OUT num_reassign INT  
)  
BEGIN  
    DECLARE done INT DEFAULT FALSE;  
    DECLARE cat_nome VARCHAR(50);  
    DECLARE default_categoria_nome VARCHAR(50) DEFAULT 'Default';  
  
    -- dichiaro il cursore per poter iterare sui figli della categoria da eliminare  
    DECLARE cur_figli CURSOR FOR  
        SELECT Nome  
        FROM categoria  
        WHERE CategoriaSuperiore IN (  
            SELECT Nome FROM TmpCategorieDaEliminare  
        );  
  
    -- handler per gestire le eccezioni  
    DECLARE EXIT HANDLER FOR SQLEXCEPTION  
    BEGIN  
        ROLLBACK;  
        RESIGNAL;  
    END;  
  
    -- handler per quando il cursore finisce i risultati elaborare  
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
```

-- creo la tabella temporanea per le categorie da eliminare

```
DROP TEMPORARY TABLE IF EXISTS TmpCategorieDaEliminare;  
CREATE TEMPORARY TABLE TmpCategorieDaEliminare (  
    Nome VARCHAR(50) PRIMARY KEY  
);
```

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;  
START TRANSACTION;
```

-- prima verifico che esista effettivamente la categoria di default per riassegnare le aste associate alle categorie cancellate

```
IF NOT EXISTS (  
    SELECT 1  
    FROM Categoria  
    WHERE Nome = default_categoria_nome  
        AND Livello = 1  
        AND CategoriaSuperiore IS NULL  
) THEN  
    SIGNAL SQLSTATE '45300'  
    SET MESSAGE_TEXT = 'Categoria di default non valida o mancante';  
END IF;
```

-- inserisce la categoria input nella tabella temporanea

```
INSERT IGNORE INTO TmpCategorieDaEliminare VALUES (nome_categoria);
```

-- loop

```
REPEAT  
    SET done = FALSE;  
  
    OPEN cur_figli;  
    read_loop: LOOP  
        FETCH cur_figli INTO cat_nome;  
        IF done THEN  
            LEAVE read_loop;
```

```
END IF;

INSERT IGNORE INTO TmpCategorieDaEliminare VALUES (cat_nome);
END LOOP;
CLOSE cur_figli;

UNTIL done END REPEAT;

-- conta quante aste verranno riassegnate prima di aggiornare
SELECT COUNT(*) INTO num_reassign
FROM asta
WHERE Categoria IN (SELECT Nome FROM TmpCategorieDaEliminare);

-- riassegna le aste a "Default"
UPDATE asta
SET Categoria = default_categoria_nome
WHERE Categoria IN (SELECT Nome FROM TmpCategorieDaEliminare);

-- elimina le categorie trovate
DELETE FROM categoria
WHERE Nome IN (SELECT Nome FROM TmpCategorieDaEliminare);

-- conta le categorie che sono state eliminate
SELECT COUNT(*) INTO num_del FROM TmpCategorieDaEliminare;

-- pulisce la tabella temporanea
DROP TEMPORARY TABLE IF EXISTS TmpCategorieDaEliminare;

COMMIT;
END
```

Utilizza una tabella temporanea e un cursore per identificare ricorsivamente tutti i figli della categoria da eliminare. Le aste associate vengono riassegnate alla categoria "Default" per mantenere



l'integrità referenziale. Il livello `SERIALIZABLE` è essenziale per evitare che altre transazioni modifichino la gerarchia durante l'eliminazione.

#### **UA4 - visualizza aste stato**

Visualizza le aste di un amministratore filtrate per stato.

##### **PROCEDURA visualizza\_aste\_stato:**

```
CREATE PROCEDURE `visualizza_aste_stato`(  
    IN var_cf CHAR(16),  
    IN var_stato VARCHAR(10)  
)  
BEGIN  
    SELECT *  
    FROM asta  
    WHERE UtenteAmministratore = var_cf AND StatoAsta = var_stato  
    ORDER BY asta.Data DESC;  
END
```

#### **UA5 - visualizza offerte asta**

Mostra le offerte di un'asta con possibilità di filtrare solo quelle automatiche. La transazione non è necessaria perché il rischio che arrivino altre offerte durante la lettura non è un problema per la visualizzazione.

##### **PROCEDURA visualizza\_offerte\_asta:**

```
CREATE PROCEDURE `visualizza_offerte_asta`(  
    IN idAsta INT,  
    IN soloAutomatiche BOOLEAN  
)  
BEGIN  
    SELECT o.UtenteBase, o.Asta, o.Data, o.Ora, o.Importo, o.Automatica  
    FROM offerta o  
    WHERE o.Asta = idAsta
```

```
AND (soloAutomatiche = FALSE OR o.Automatica = 1)
ORDER BY o.Data, o.Ora;
```

```
END
```

### **Operazioni Utente Base (UB)**

#### **UB1, UB2, UB3 - visualizzaAsteFiltrate**

Unifica tre operazioni simili attraverso parametri opzionali: visualizza aste attive (UB1), per categoria (UB2) o per amministratore (UB3). L'utilizzo della vista vista\_aste\_attive ottimizza le performance. Utilizzo READ COMMITTED come livello di isolamento perché ho bisogno che le aste siano state effettivamente create per non visualizzare dati inconsistenti, ma non mi interessa visualizzare ogni aggiornamento in tempo reale.

#### **PROCEDURA visualizzaAsteFiltrate:**

```
CREATE PROCEDURE `visualizzaAsteFiltrate`(  
    IN var_categoria VARCHAR(50),  
    IN var_amministratore CHAR(16)  
)  
BEGIN  
    SET TRANSACTION ISOLATION LEVEL READ COMMITTED;  
    START TRANSACTION;  
  
    SELECT *  
    FROM vista_aste_attive  
    WHERE  
        (var_categoria IS NULL OR Categoria = var_categoria)  
        AND (var_amministratore IS NULL OR UtenteAmministratore = var_amministratore)  
    ORDER BY TempoRimanenteSec;  
  
    COMMIT;  
END
```

**UB4 - asteVinteUtente**

Semplice query per visualizzare gli articoli (aste) aggiudicati da un utente. Non richiede transazione esplicita.

**PROCEDURA asteVinteUtente:**

```
CREATE PROCEDURE `asteVinteUtente`(  
    IN var_cf CHAR(16)  
)  
BEGIN  
    SELECT ID, Descrizione, Dimensioni, CondizioniArticolo, Categoria, PrezzoBase,  
    OffertaMassima, Data, Durata, NumOfferte, UtenteAmministratore  
    FROM asta  
    WHERE UtenteBase = var_cf AND StatoAsta = 'TERMINATA';  
END
```

**UB5 - aggiungiOfferta**

Procedura critica che richiede particolare attenzione transazionale, infatti è stato scelto **SERIALIZABLE** come livello di isolamento per prevenire che altri utenti diventino migliori offerenti durante il controllo. Inoltre, viene controllato che l'utente non sia già il miglior offerente. L'inserimento dell'offerta e l'aggiornamento delle ridondanze (OffertaMassima, NumOfferte) devono essere atomici e l'importo massimo per controfferte automatiche viene registrato per future elaborazioni da parte del sistema di controfferta automatica (rilanci automatici).

La complessità transazionale è giustificata dalla necessità di mantenere la consistenza in un ambiente altamente concorrente dove multiple offerte possono arrivare simultaneamente da più utenti diversi.

**PROCEDURA aggiungiOfferta:**

```
CREATE PROCEDURE `aggiungiOfferta`(  
    IN var_utenteBase CHAR(16),  
    IN var_idAsta INT,  
    IN var_importo DECIMAL(7,2),  
    IN var_automatica BOOLEAN,
```

```
IN var_maxControfferta DECIMAL(7,2)
)
BEGIN
    DECLARE incremento DECIMAL(7,2) DEFAULT 0.50;
    DECLARE miglior_attuale CHAR(16);
    DECLARE offerta_massima DECIMAL(7,2);

    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
    START TRANSACTION;

    -- controllo se è già il miglior offerente
    SELECT UtenteBase INTO miglior_attuale
    FROM asta
    WHERE ID = var_idAsta;

    IF miglior_attuale = var_utenteBase THEN
        SIGNAL SQLSTATE '45900'
        SET MESSAGE_TEXT = 'Sei già il miglior offerente e non puoi rilanciare.';
    END IF;

    -- inserimento dell'offerta
    INSERT INTO offerta (UtenteBase, Asta, Data, Ora, Importo, Automatica, ImportoControfferta)
    VALUES (var_utenteBase, var_idAsta, CURDATE(), CURRENT_TIME(6), var_importo,
    var_automatica, var_maxControfferta);

    COMMIT;
END
```

**UB6 - visualizzaAstePartecipate**

Procedura che mostra le aste a cui l'utente sta partecipando con il dettaglio dell'ultima offerta fatta. Utilizza subquery per identificare l'offerta più recente per ogni asta. La transazione READ COMMITTED garantisce una vista consistente dei dati.

**PROCEDURA visualizzaAstePartecipate:**

```
CREATE PROCEDURE `visualizzaAstePartecipate`(  
    IN var_cf CHAR(16)  
)  
BEGIN  
    SET TRANSACTION ISOLATION LEVEL READ COMMITTED;  
    START TRANSACTION;  
  
    SELECT  
        va.*,  
        o.Importo AS ImportoUltimaOfferta,  
        o.Data AS DataUltimaOfferta,  
        o.Ora AS OraUltimaOfferta,  
        o.Automatica AS ControffertaAttiva,  
        o.ImportoControfferta AS ImportoControfferta  
    FROM vista_aste_attive va  
    JOIN (  
        SELECT o.*  
        FROM offerta o  
        INNER JOIN (  
            SELECT Asta, MAX(CONCAT(Data, ' ', Ora)) AS MaxDateTime  
            FROM offerta  
            WHERE UtenteBase = var_cf  
            GROUP BY Asta  
        ) ultimaOff  
        ON o.Asta = ultimaOff.Asta AND CONCAT(o.Data, ' ', o.Ora) = ultimaOff.MaxDateTime  
        WHERE o.UtenteBase = var_cf
```

```
) o ON va.ID = o.Asta
```

```
ORDER BY va.TempoRimanenteSec;
```

```
COMMIT;
```

```
END
```

### **Procedure di supporto**

#### **registerUser** - Registrazione nuovo utente base

Gestisce l'inserimento atomico su tre tabelle (cartadicredito, utentebase, login) con controlli su lunghezza username/password, hash SHA2-256 per password e numero carta e messaggi personalizzati per violazioni di unicità. Il livello di isolamento READ COMMITTED è necessario per evitare dirty reads e quindi evitare di inserire uno username scelto da un altro utente che si sta registrando.

#### **PROCEDURA registerUser:**

```
CREATE PROCEDURE `registerUser`(  
    in var_username VARCHAR(50),  
    in var_password VARCHAR(32),  
    in var_cf CHAR(16),  
    in var_nome VARCHAR(30),  
    in var_cognome VARCHAR(30),  
    in var_indirizzo VARCHAR(45),  
    in var_cap CHAR(5),  
    in var_citta VARCHAR(45),  
    in var_data_nascita DATE,  
    in var_citta_nascita VARCHAR(45),  
    in var_carta_credito VARCHAR(16),  
    in var_cvv CHAR(3),  
    in var_data_scadenza_carta DATE  
)  
BEGIN  
    DECLARE errno INT DEFAULT 0;  
    DECLARE msg TEXT;  
    DECLARE hashed_cc CHAR(64);
```

```
DECLARE hashed_password CHAR(64);
```

```
DECLARE EXIT HANDLER FOR SQLEXCEPTION
```

```
BEGIN
```

```
  GET DIAGNOSTICS CONDITION 1
```

```
    errno = MYSQL_ERRNO,
```

```
    msg = MESSAGE_TEXT;
```

```
  ROLLBACK;
```

```
  -- Gestione errore di chiave duplicata
```

```
  IF errno = 1062 THEN
```

```
    IF msg LIKE '%login.UNIQUE_username%' OR msg LIKE '%login.PRIMARY%' THEN
```

```
      SIGNAL SQLSTATE '45001' SET MESSAGE_TEXT = 'Errore: username esistente',
```

```
      MYSQL_ERRNO = 1062;
```

```
    ELSEIF msg LIKE '%utentebase.UNIQUE_cf%' OR msg LIKE '%utentebase.PRIMARY%'
```

```
    THEN
```

```
      SIGNAL SQLSTATE '45002' SET MESSAGE_TEXT = 'Errore: codice fiscale registrato',
```

```
      MYSQL_ERRNO = 1062;
```

```
    ELSEIF msg LIKE '%cartadicredito.UNIQUE_numero%' OR msg LIKE  
'%cartadicredito.PRIMARY%' THEN
```

```
      SIGNAL SQLSTATE '45003' SET MESSAGE_TEXT = 'Errore: carta di credito presente',
```

```
      MYSQL_ERRNO = 1062;
```

```
    ELSE
```

```
      SIGNAL SQLSTATE '45004' SET MESSAGE_TEXT = 'Errore: valore duplicato',
```

```
      MYSQL_ERRNO = 1062;
```

```
    END IF;
```

```
  ELSE
```

```
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg, MYSQL_ERRNO = errno;
```

```
  END IF;
```

```
END;
```

```
SET hashed_cc = sha2(var_carta_credito, 256);
```

```
SET hashed_password = sha2(var_password, 256);
```

SET TRANSACTION ISOLATION LEVEL READ COMMITTED; -- Si vuole escludere la possibilità di violazioni del tipo lettura sporca (dirty reads)

START TRANSACTION;

-- validazioni da effettuare sui dati in ingresso

IF CHAR\_LENGTH(var\_username) < 4 THEN

SIGNAL SQLSTATE '45005' SET MESSAGE\_TEXT = 'Username troppo corto';

END IF;

IF CHAR\_LENGTH(var\_password) < 8 THEN

SIGNAL SQLSTATE '45006' SET MESSAGE\_TEXT = 'Password troppo corta';

END IF;

INSERT INTO cartadicredito (

Numero, CVV, DataScadenza

) VALUES (

hashed\_cc, var\_cvv, var\_data\_scadenza\_carta

);

INSERT INTO utentebase (

CF, Nome, Cognome, Indirizzo, CAP, Città, DataNascita, CittàNascita, CartaCredito

) VALUES (

var\_cf, var\_nome, var\_cognome, var\_indirizzo, var\_cap, var\_citta, var\_data\_nascita,

var\_citta\_nascita, hashed\_cc

);

INSERT INTO login (

username, password, ruolo, CF

) VALUES (

var\_username, hashed\_password, 'USER', var\_cf

);

COMMIT;

END



**aggiorna indirizzo consegna - Modifica dati spedizione**

Procedura con verifica del successo attraverso ROW\_COUNT(). Effettua ROLLBACK se nessuna riga viene aggiornata. Questa procedura serve per modificare l'indirizzo di consegna inserito all'atto di registrazione. Il livello di isolamento READ COMMITTED è necessario dato il controllo esplicito tramite ROW\_COUNT().

**PROCEDURA aggiorna\_indirizzo\_consegna:**

```
CREATE PROCEDURE `aggiorna_indirizzo_consegna`(  
    in var_cf CHAR(16),  
    in var_indirizzo VARCHAR(45),  
    in var_citta VARCHAR(45),  
    in var_cap CHAR(5),  
    out result INT  
)  
BEGIN  
    declare righe_aggiornate int;  
    declare exit handler for sqlexception  
    begin  
        rollback;  
        set result = 0;  
    end;  
  
    SET TRANSACTION ISOLATION LEVEL READ COMMITTED;  
    START TRANSACTION;  
  
    update utentebase  
    set  
        Indirizzo = var_indirizzo,  
        Città = var_citta,  
        CAP = var_cap  
    where CF = var_cf;
```

```
set righe_aggiornate = row_count();

if righe_aggiornate > 0 then
    commit;
    set result = 1; -- aggiornamento riuscito
else
    rollback;
    set result = 0; -- nessun utente trovato con quel CF
end if;

END
```

### **chiudiAsta** - Chiusura manuale di un'asta attiva

Permette all'amministratore di terminare anticipatamente un'asta attiva. Utilizza READ COMMITTED per garantire che lo stato dell'asta non venga modificato da altre transazioni durante il controllo e l'aggiornamento.

#### **PROCEDURA chiudiAsta:**

```
CREATE PROCEDURE `chiudiAsta`(  
    IN var_idAsta INT  
)  
BEGIN  
    SET TRANSACTION ISOLATION LEVEL READ COMMITTED;  
    START TRANSACTION;  
  
    UPDATE asta  
    SET StatoAsta = 'TERMINATA'  
    WHERE ID = var_idAsta AND StatoAsta = 'ATTIVA';  
  
    COMMIT;  
END
```

### **dettagli utente cf** - Recupero dettagli utente per autenticazione

Procedura di supporto utilizzata dopo il login per recuperare i dettagli completi dell'utente dalla tabella appropriata (UtenteBase o UtenteAmministratore) in base al ruolo.

PROCEDURA dettagli\_utente\_cf:

```
CREATE PROCEDURE `dettagli_utente_cf`(  
    in var_cf CHAR(16),  
    in var_ruolo ENUM('ADMIN', 'USER')  
)  
BEGIN  
    if var_ruolo = 'USER' then  
        select *  
        from UtenteBase  
        where CF = var_cf;  
    else  
        select *  
        from UtenteAmministratore  
        where CF = var_cf;  
    end if;  
END
```

**getControfferteAttive** - Recupero controfferte da processare

Query complessa che identifica le controfferte automatiche attivabili quando viene superata l'offerta corrente. Utilizza una subquery per trovare l'importo massimo di controfferta per ogni utente, escludendo il miglior offerente attuale. Utilizza `SERIALIZABLE` per garantire isolamento completo durante il processo di rilanci automatici: deve operare sullo stesso snapshot di dati di `getMigliorOfferta` per prevenire inconsistenze nel loop di controfferte, bloccando inserimenti concorrenti che potrebbero alterare la logica di rilancio.

PROCEDURA getControfferteAttive:

```
CREATE PROCEDURE `getControfferteAttive`(  
    IN var_idAsta INT,  
    IN var_migliorOfferente CHAR(16),
```

```
IN var_importoAttuale DECIMAL(7,2)
)
BEGIN
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
START TRANSACTION;

SELECT o.*
FROM offerta o
INNER JOIN (
    SELECT UtenteBase, MAX(ImportoControfferta) as MaxImporto
    FROM offerta
    WHERE Asta = var_idAsta
    AND ImportoControfferta IS NOT NULL
    AND ImportoControfferta > var_importoAttuale
    AND UtenteBase <> var_migliorOfferente
    GROUP BY UtenteBase
) AS maxOfferte ON o.UtenteBase = maxOfferte.UtenteBase
    AND o.ImportoControfferta = maxOfferte.MaxImporto
    AND o.Asta = var_idAsta
ORDER BY o.ImportoControfferta DESC;

COMMIT;
END
```

### **getMigliorOfferta - Identificazione offerta vincente**

Determina l'offerta vincente corrente ordinando per importo e, in caso di parità, per data/ora (FIFO). Essenziale per il sistema di controfferte e per determinare il vincitore alla chiusura. Query di sola lettura senza necessità di transazione. Utilizza SERIALIZABLE per garantire isolamento completo durante l'elaborazione delle controfferte automatiche, impedendo inserimenti di nuove offerte concorrenti, assicurando che il sistema di rilanci automatici operi su una lista di dati completamente stabile dell'asta.

### **PROCEDURA getMigliorOfferta:**

```
CREATE PROCEDURE `getMigliorOfferta`(  
    IN var_idAsta INT  
)  
BEGIN  
    SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;  
    START TRANSACTION;  
  
    SELECT *  
    FROM offerta  
    WHERE Asta = var_idAsta  
    ORDER BY Importo DESC, Data DESC, Ora DESC  
    LIMIT 1;  
  
    COMMIT;  
END
```

### **listaAmministratori** - Elenco amministratori del sistema

Restituisce l'elenco degli amministratori per popolare filtri di ricerca (operazione UB3).

#### **PROCEDURA listaAmministratori:**

```
CREATE PROCEDURE `listaAmministratori`()  
BEGIN  
    SELECT CF, Nome, Cognome  
    FROM utenteamministratore;  
END
```

### **listaCategorie** - Elenco completo delle categorie

Fornisce la struttura gerarchica delle categorie per i menu di selezione. Utilizzata sia da amministratori che utenti base. Utilizza REPEATABLE READ per garantire una vista consistente e immutabile della gerarchia durante la lettura, impedendo che modifiche concorrenti alterino la struttura mentre viene costruito il menu delle categorie, lasciando comunque possibili gli inserimenti di nuove categorie.

PROCEDURA listaCategorie:

```
CREATE PROCEDURE `listaCategorie`()
BEGIN
    SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
    START TRANSACTION;

    SELECT Nome, Livello, CategoriaSuperiore
    FROM categoria;

    COMMIT;
END
```