|  |  |
| --- | --- |
|  |  |

Basi di Dati

Progetto A.A. 2024/2025

ASTE ONLINE

0281017

Alessandro Pica

**Indice**

[1. Descrizione del Minimondo 2](#_Toc527814491)

[2. Analisi dei Requisiti 3](#_Toc902236831)

[3. Progettazione concettuale 4](#_Toc55390959)

[4. Progettazione logica 5](#_Toc180212209)

[5. Progettazione fisica 7](#_Toc1804305735)

# Descrizione del Minimondo

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | 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€.  Il sistema tiene traccia, per ogni oggetto, di tutte le offerte che sono state fatte e dell’instante 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, in ogni momento, possono visualizzare l’elenco degli oggetti aggiudicati e l’elenco degli oggetti per i quali è presente un’asta in corso cui hanno fatto almeno un’offerta |

# Analisi dei Requisiti

## Identificazione dei termini ambigui e correzioni possibili

|  |  |  |  |
| --- | --- | --- | --- |
| **Linea** | **Termine** | **Nuovo termine** | **Motivo correzione** |
| 3 | Utenti del sistema | Clienti | I clienti sono le persone che effettuano offerte alle aste; per utenti del sistema si intendono tutti gli utenti registrati (quindi sia amministratori sia clienti) |
| 4 | Oggetto | Asta | Le offerte vengono effettuate sulle aste. |
| 14 | Asta | Oggetto | Le categorie sono degli oggetti non dell’asta. |
| 31 | Transazioni | Offerte | Il sistema “controfferta automatica” genera offerte. |

### Specifica disambiguata

|  |
| --- |
| Una casa d’aste intende realizzare un sistema online di aste. Il sistema deve consentire agli amministratori la gestione degli oggetti che si vogliono pubblicare e tutto il ciclo di vita delle aste. I clienti previa registrazione hanno la possibilità di fare offerte su una qualsiasi asta. Al termine dell’asta, l’offerta maggiore sarà quella che avrà vinto l’asta. Alla registrazione, i clienti 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 ciascun oggetto viene associato 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.  I clienti possono visualizzare in qualsiasi momento tutte le aste aperte. Quando un’asta viene visualizzata, i clienti 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, i clienti 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 cliente che ha attualmente piazzato l’offerta massima, può sfruttare la funzionalità di “controfferta automatica”. Tale funzionalità permette al cliente di indicare un importo massimo con cui si intende rilanciare l’offerta, qualora un altro cliente faccia un’offerta maggiore. La gestione delle offerte pertanto funziona nel modo seguente. Il cliente A indica un importo I con cui vuole rilanciare l’offerta nei confronti del cliente B che è attualmente il migliore offerente. Il cliente B ha anche indicato un importo di controfferta C. Se C > I, il sistema indicherà come miglior offerente il cliente A, con importo temporaneo I, ma immediatamente dopo indicherà nuovamente il cliente B come migliore offerente, con un importo di I + 0,50€.  Il sistema tiene traccia, per ogni oggetto, di tutte le offerte che sono state fatte e dell’instante temporale in cui queste sono state inserite nel sistema. Ciò significa che tutte le offerte automatiche generate dal sistema di controfferta automatica devono essere registrate nel sistema.  I clienti, in ogni momento, possono visualizzare l’elenco degli oggetti aggiudicati e l’elenco degli oggetti per i quali è presente un’asta in corso cui hanno fatto almeno un’offerta. |

## Glossario dei Termini

|  |  |  |  |
| --- | --- | --- | --- |
| **Termine** | **Descrizione** | **Sinonimi** | **Collegamenti** |
| Cliente | Persona che a seguito di una registrazione, effettua offerte per poter comprare un oggetto | Offerente | Aste, Oggetti. |
| Amministratore | Persona che a seguito di una registrazione, ha la responsabilità di creare categorie ed aste |  | Categorie,Aste, Oggetti |
| Oggetti | Prodotti della casa d’aste destinata alla vendita. |  | Amministratore, Clienti,Aste, Categorie. |
| Offerta | Proposta economica relativa ad una singola asta per comprare un oggetto |  | Oggetti, Clienti. |

## Raggruppamento dei requisiti in insiemi omogenei

|  |
| --- |
| **Frasi relative a Cliente** |
| - I clienti, previa registrazione, hanno la possibilità di fare offerte su una qualsiasi asta.  - Alla registrazione, i clienti 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;  - i clienti possono visualizzare in qualsiasi momento tutte le aste aperte.  -Quando un’asta viene visualizzata, i clienti 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;  - i clienti possono fare un’offerta, maggiore del valore attuale di offerta;  - il cliente che ha attualmente piazzato l’offerta massima, può sfruttare la funzionalità di “controfferta automatica”;  - i clienti, in ogni momento, possono visualizzare l’elenco degli oggetti aggiudicati e l’elenco degli oggetti per i quali è presente un’asta in corso cui hanno fatto almeno un’offerta. |
| **Frasi relative a 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 ciascun oggetto viene associato una categoria. |
| **Frasi relative a Offerte** |
| -il cliente che ha attualmente piazzato l’offerta massima, può sfruttare la funzionalità di “controfferta automatica”. Tale funzionalità permette al cliente di indicare un importo massimo con cui si intende rilanciare l’offerta, qualora un altro cliente faccia un’offerta maggiore. La gestione delle offerte pertanto funziona nel modo seguente. Il cliente A indica un importo I con cui vuole rilanciare l’offerta nei confronti del cliente B che è attualmente il migliore offerente. Il cliente B ha anche indicato un importo di controfferta C. Se C > I, il sistema indicherà come miglior offerente il cliente A, con importo temporaneo I, ma immediatamente dopo indicherà nuovamente il cliente B come migliore offerente, con un importo di I + 0,50€.  -Il sistema tiene traccia, per ogni oggetto, di tutte le offerte che sono state fatte e dell’instante temporale in cui queste sono state inserite nel sistema.  -Tutte le offerte automatiche generate dal sistema di controfferta automatica devono essere registrate nel sistema. |

# Progettazione concettuale

## Costruzione dello schema E-R

Per quanto riguarda la costruzione dello schema E-R si è deciso di utilizzare la strategia top-down, che ha diverse primitive di trasformazione, iniziamo dapprima introducendo le entità in gioco:

* Asta:
* Offerte
* Oggetto
* Categoria
* Amministratore
* Cliente

successivamente, c’è stata una fase di definizione delle relazioni tra queste entità e sono emerse le seguenti relazioni tra le entità:

* Un OGGETTO APPARTIENE ad una CATEGORIA (Figura 1);
* Un AMMINISTRATORE CREA un OGGETTO(Figura2);
* Un CLIENTE POSSIEDE un OGGETTO che si è aggiudicato (Figura3);
* Un CLIENTE FORMULA un OFFERTA(Figura4);
* Un OGGETTO è RELATIVO ad un ASTA(Figura5);
* Un’OFFERTA fa RIFERIMENTO ad un’ASTA(Figura6);

dopodiché si è riesaminata la specifica e si sono trovati tutti gli attributi relativi per ogni entità:

* AMMINISTRATORE (NomeUtente, Password);
* CLIENTE (NomeUtente, Password, CF, Nome, Cognome, DataNascita, CittàNascita);
* OGGETTO (Codice, Descrizione, PrezzoBase, Dimensioni,Stato);
* ASTA (Oggetto, Durata, DataInizio);
* OFFERTE (OffertaMassima, ValoreEffettivo);
* CATEGORIA(NomeCategoria).

Dopo tutte queste varie fasi, si sono prodotti i seguenti sottoschemi ER:

Immagine che contiene diagramma, testo, Piano, Disegno tecnico

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 1: relazione tra Categoria ed Oggetto.

Immagine che contiene diagramma, testo, schizzo, disegno

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 2: Relazione tra Cliente e Oggetto

Immagine che contiene diagramma, testo, Piano, linea

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 3: relazione tra Cliente e Offerta

L’ultima fase prevede l’integrazione di tutti questi schemi ER in un unico schema ER:

Immagine che contiene diagramma, schizzo, testo, Disegno tecnico

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 7: schema ER finale

### Integrazione finale

Immagine che contiene diagramma, schizzo, testo, Disegno tecnico

Il contenuto generato dall'IA potrebbe non essere corretto.

## Regole aziendali

* Un utente che acquista un oggetto DEVE aver fatto l’offerta vincente su quell’oggetto in asta;
* Un utente con la carta di credito scaduta NON DEVE poter acquistare un oggetto;
* La durata dell’asta DEVE essere compresa tra 1 e 7 giorni;
* Le categorie DEVONO essere organizzate su un massimo di tre livelli;
* Il numero di offerte fatte SI OTTIENE sommando il numero di offerte fatte su quell’oggetto;
* L’importo di un’offerta DEVE essere maggiore del valore effettivo attuale.
* La granularità di incremento delle offerte DEVE essere multipla di 50 centesimi di euro;
* Il tempo mancante per la chiusura dell’asta SI OTTIENE dalla data di inizio dell’asta e dalla durata;
* Il tempo mancante per la chiusura dell’asta SI OTTIENE dalla data di terminazione dell’asta e dalla data odierna;
* Al termine della relativa asta, le offerte DEVEVONO essere cancellate.

## Dizionario dei dati

|  |  |  |  |
| --- | --- | --- | --- |
| **Entità** | **Descrizione** | **Attributi** | **Identificatori** |
| Cliente | È colui che piazza le offerte per concorrere all’acquisto di un oggetto | Nome, Cognome, CF,  Data di nascita, Città di nascita, Password, Nome Utente | NomeUtente |
| Oggetto in vendita | Oggetto attualmente in asta | Codice, Descrizione,  prezzo base, lunghezza, larghezza, altezza, importo offerta massima, numero offerte, tempo mancante, stato, data, orario | Codice |
| Offerte | Proposte realizzate dai clienti | Asta, Cliente, Valore effettivo, offerta massima | Oggetto in vendita, Cliente, Valore effettivo |
| Oggetto venduto | Oggetto acquisito a seguito di un’asta da un cliente | Prezzo di vendita | Codice |
| Oggetto non venduto | Oggetto rimasto a carico dell’agenzia d’aste a seguito della scadenza della relativa asta |  | Codice |
| Categoria | Classe di un singolo oggetto | Nome categoria | Nome categoria |

|  |  |  |  |
| --- | --- | --- | --- |
| **Relazione** | **Descrizione** | **Entità Coinvolte** | **Attributi** |
| Formulare | Associa Utente ad Offerta | Utente (0, N), Offerta (1, 1) |  |
| Possiede | Associa un Utente all’Oggetto Venduto che acquista | Utente (0, N),  Oggetto Venduto (1, 1) |  |
| Appartenenza | Associa Offerta ad Oggetto in Asta | Offerta (1, 1), Oggetto in Asta (0, N) |  |
| Assegnato | Associa Oggetto in Asta ad una Categoria | Oggetto in Asta (1, 1), Categoria (0, N) |  |
| Gerarchia | Associa una Categoria con le eventuali Categorie | Macrocategoria (0, 1), Sottocategoria (0, N) |  |

# Progettazione logica

## Volume dei dati.

Al fine del calcolo dei vari valori attesi si sono supposte le seguenti ipotesi:

* Nell’analisi dei volumi si considera che i dati relativi al ciclo di vita delle aste vengano mantenuti dal sistema per un periodo pari a cinque anni.
* I dati di seguito sono calcolati per un anno di tempo.
* Si suppone che il numero di Cliente che utilizzino il sistema è pari a 1.500.
* Il numero delle aste è pari a 2.000.
* Il tasso di partecipazione da parte degli utenti alle aste è del 10%, di cui il 5% selezionerà il meccanismo facoltativo dell’offerta automatica.
* In media avrò un numero di offerte ad asta pari a 10.
* Avrò quindi 16.000 offerte totali.
* Il numero di oggetti venduti è circa l’80 % quindi sarà di 1600 oggetti venduti.
* Il numero di oggetti non venduti è di 400.
* Il numero di categorie è 30.

|  |  |  |
| --- | --- | --- |
| **Concetto nello schema** | **Tipo[[1]](#footnote-1)** | **Volume atteso** |
| Utente | E | 1500 |
| Oggetto in vendita | E | 2000 |
| Oggetto venduto | E | 1600 |
| Oggetto non venduto | E | 400 |
| Offerta | E | 16000 (800 controfferte automatiche) |
| Categoria | E | 30 |
| Appartenenza | R | 1600 |
| Posseduto | R | 1600 |
| Riferimento | R | 16000 |
| Gerarchia | R | 30 |
| Formulare | R | 16000 |

## Tavola delle operazioni

|  |  |  |
| --- | --- | --- |
| **Cod.** | **Descrizione** | **Frequenza attesa** |
| L1 | Registrazione | 1500 |
| L2 | Login | 185/giorno |
| C1 | Realizzare un’offerta | 420/giorno |
| C2 | Offerta automatica | 45/giorno |
| C3 | Visualizza aste aperte[[2]](#footnote-2) | 185/giorno |
| C4 | Visualizza elenco oggetti aggiudicati | 10/mese |
| C5 | Visualizza aste in corso[[3]](#footnote-3) | 185/giorno |
| AM1 | Inserimento asta | 5/giorno |
| AM2 | Creazione Categoria | 5/anno |
| S1 | Backup Dati | 1/anno |
| S2 | Cancellazione dati | 1/ 5 anni |

## Costo delle operazioni

Come ipotesi, supponiamo che l’accesso in scrittura costi il doppio rispetto ad un accesso in sola lettura.

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Cliente | E | 1 | L |
| Cliente | E | 2 | S |

Tabella 1: operazioni L1(Login) e operazioni L2(Registrazione)

Stimando circa 185 accessi giornalieri e costo per singolo accesso pari a 1, il costo stimato per l’operazione di login è di 185 accessi.

Stimando circa 300 registrazioni annuali e supponendo il costo per singola registrazione pari a 2, il costo stimato annualmente per la registrazione è di 600 accessi.

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Cliente | E | 1 | L |
| Oggetto in asta | E | 1 | L |
| Riferimento | R | 10 | L |
| Offerta | E | 2 | S |
| Controfferta automatica | E | 1 | L |
| Formulare | R | 2 | S |
| Cliente | E | 2 | S |
| Controfferta automatica | E | 2 | S |
| Riferimento | R | 2 | S |

Tabella 2: operazione C1 (Realizzare offerta)

Per realizzare un’offerta abbiamo un costo totale di 23 per singola offerta; avendo supposto 420 offerte giornaliere otteniamo 9660 accessi

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Cliente | E | 1 | L |
| Formulare | R | 1 | L |
| Offerta | E | 1 | L |
| Riferimento | E | 1 | L |
| Oggetto In Vendita | E | 2 | S |
| Offerte | E | 2 | S |

Tabella 3: operazione C2 (Offerta Automatica)

Avrò circa 45 controfferte automatiche al giorno e al costo di 8 accessi unitari, ho il costo totale giornaliero di accessi pari a 360 accessi.

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Oggetto in asta | E | 1 | L |
| Appartenenza | R | 1 | L |
| Categoria | E | 1 | L |
| Gerarchia | R | 1 | L |

Tabella 4: operazione C3 visualizza aste aperte

In media verrò 2 categorie e avrò circa 185 accessi per vedere le aste aperte; per un costo totale di 925 accessi

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Cliente | E | 1 | L |
| Posseduto | R | 1 | L |
| Oggetto In Vendita | E | 1 | L |

Tabella 5: operazione C4 visualizza oggetti aggiudicati

Qui ho un costo di 3 accessi per circa 10 volte al mese quindi avrò un costo di 30 accessi al mese.

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Cliente | E | 1 | L |
| Formulare | R | 1 | L |
| Offerte | E | 1 | L |
| Riferimento | R | 1 | L |
| Oggetto in asta | E | 1 | L |
| Categoria | E | 1 | L |
| Gerarchia | E | 1 | L |

Tabella 6: operazione C5 visualizza aste in corso

In media avrò 2 categorie e circa 185 accessi per vedere le aste in corso, la singola operazione costa 8 accessi per un totale di 1480 accessi.

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Oggetto in Vendita | E | 2 | S |
| Appartenenza | R | 2 | S |
| Categoria | E | 2 | S |
| Gerarchia | R | 2 | S |
| Oggetto in asta | E | 1 | L |
| Categoria | E | 1 | L |
| Gerarchia | E | 1 | L |

Tabella 7: Offerta AM1 Aggiungi Oggetto in Asta

In media avrò 2 categorie ad oggetto; e mi aspetto di avere 5 oggetti aggiunti al giorno; il singolo costo è di 12 accessi per un costo totale di 60 accessi giornalieri.

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Categoria | E | 2 | S |
| Gerarchia | R | 2 | S |

Tabella 8 : Offerta AM2 Crea Categoria

Supponendo in media 2 categorie ad oggetto, mi aspetto di creare nuove categorie 5 volte l’anno per un costo totale di 20 accessi l’anno.

## Ristrutturazione dello schema E-R

**Analisi delle ridondanze:**

*Numero Offerte e Valore Massima Offerta*

Il numero di offerte dell’entità Oggetto in Asta rappresenta una ridondanza, in quanto è ricavabile tramite il conteggio delle occorrenze delle entità Offerte riferite a quello specifico oggetto, navigando opportunamente la relazione Riferimento. Anche il dato relativo al valore della massima offerta valida per un preciso Oggetto in asta rappresenta una ridondanza, in quanto basterebbe vedere l’ultima offerta valida effettuata per un dato oggetto.

Le operazioni che operano, in qualche modo, sul numero di offerte sono:

* Operazione U1 – Fare un’offerta;
* Operazione U2 – Fare un’offerta impostando la controfferta automatica;
* Operazione U3 – Visualizza Aste Aperte

Il costo delle operazioni senza la ridondanza è riportato nelle tabelle relative discusse in precedenza.

Consideriamo per semplicità il caso in cui un utente generi un’offerta manualmente, ma lo stesso ragionamento può essere fatto con il caso in cui l’offerta venga generata impostando la funzionalità Controfferta Automatica poiché, come abbiamo visto in precedenza, il costo delle operazioni in termini di accessi rimane invariato.

Supponiamo che il numero di offerte sia mantenuto nella base di dati come un tipo di dato intero occupante 4 B e quindi, considerando la tavola dei volumi sopra riportata, abbiamo un totale di

4 B \* 10000 Oggetto in Asta = 40.000 B = 40 kB

di memoria utilizzata.

Supponendo che il valore della massima offerta venga mantenuto nella base di dati con un tipo di dato in virgola mobile occupante 8 B e, considerando la tavola dei volumi sopra riportata, abbiamo un totale di

8 B \* 10.000 Oggetto in Asta = 80.000 B = 8 kB

Di memoria utilizzata.

Considerando entrambe le ridondanze abbiamo un totale di 48 kB di memoria utilizzata nella base di dati.

Nelle seguenti tabelle viene indicato il numero e il tipo degli accessi in presenza delle ridondanze.

## Traduzione di entità e associazioni

Partendo dallo schema E-R ristrutturato, raffigurato poco sopra, costruiamo le seguenti relazioni per il modello relazionale.

• *Cliente*(CF, Nome, Cognome, Data di Nascita, Città di Nascita, Numero Carta di Credito, Data di Scadenza Carta di Credito, CVV, Via di Consegna, Numero Civico di Consegna, Città, CAP di Consegna)

• *Oggetto in Asta* (Codice, Descrizione, Stato, Prezzo di Base, Descrizione Larghezza,Altezza,Lunghezza, , Numero Offerte\*, Valore Massima Offerta\*, Data Inizio Asta,Orario Inizio Asta , Durata Asta, ,Categoria, Utente\*)

o *Oggetto in Asta*(Categoria) ⊆ *Categoria*(Nome)

o *Oggetto in Asta*(Utente) ⊆ *Utente*(CF)

• *Offerta*(Utente, Importo, Oggetto In Asta, , Controfferta massima\*)

* o *Offerta*(Utente) ⊆ *Utente*(CF)
* o *Offerta*(Oggetto In Asta) ⊆ *Oggetto In Asta*(Codice)
* • *Categoria*(Nome, Macrocategoria\*)
* o *Categoria*(Macrocategoria*)* ⊆ *Categoria*(Nome)

Nota: \* indica che l’attributo può avere valori NULL

## Normalizzazione del modello relazionale

1NF

Ricordando che una relazione soddisfa la 1NF se gli attributi della relazione sono definiti su valori atomici e non su valori complessi quali insiemi o relazioni; pertanto, si ha che tutte le relazioni sono in prima forma normale

2NF

Una relazione è in seconda forma normale se è in prima forma normale e su di essa non sono definite dipendenze parziali, cioè dipendenze fra un sottoinsieme proprio della chiave e altri attributi; pertanto, si ha che le relazioni sono in seconda forma normale

3NF

Una relazione è in terza forma normale se è in seconda forma normale e su di essa non sono presenti dipendenze transitive. Dato che le relazioni sono in seconda forma normale e non sono presente dipendenze transitive, le relazioni sono in seconda forma normale.

# Progettazione fisica

## Utenti e privilegi

## Le tipologie di utenti e i relativi privilegi all’interno del sistema sono:

* Login:
  + Grant in esecuzione sulle operazioni L1 ed L2.
* Utente:
  + Grant in esecuzione sulle operazioni C1, C2, C3, C4, C5.
* Amministratori:
  + Grant in esecuzione sulle operazioni AM1, AM2,.

Si osservi che, per far sì che ciò avvenga, abbiamo bisogno di aggiungere una entità relativa agli Amministratori per gestire gli amministratori all’interno del sistema, che avrà come attributi l’username e la password dell’amministratore; aggiungiamo all’entità Utente gli attributi di username e password-

## Strutture di memorizzazione

|  |  |  |
| --- | --- | --- |
| **Tabella Categoria** | | |
| **Colonna** | **Tipo di dato** | **Attributi[[4]](#footnote-4)** |
| **Nome Categoria** | VARCHAR(30) | PK,NN |
| **Categoria Superiore** | VARCHAR(45) |  |

|  |  |  |
| --- | --- | --- |
| **Tabella Controfferte in coda** | | |
| **Colonna** | **Tipo di dato** | **Attributi[[5]](#footnote-5)** |
| **Cliente** | VARCHAR(30) | NN |
| **ID Coda** | INT | NN,AI,PK |
| **Controfferta Massima** | DOUBLE | NN |
| **Data Richiesta** | DATETIME | NN |
| **Messaggio Errore** | TEXT |  |
| **Stato** | ENUM | NN |
| **Oggetto in vendita** | VARCHAR(5) | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella Offerte** | | |
| **Colonna** | **Tipo di dato** | **Attributi[[6]](#footnote-6)** |
| **Cliente** | VARCHAR(30) | PK,NN |
| **ControffertaMassima** | DOUBLE | NN |
| **OggettoInVendita** | VARCHAR(5) | PK,NN |
| **ValoreEffettivo** | DOUBLE | PK,NN |

|  |  |  |
| --- | --- | --- |
| **Tabella OggettiInAsta** | | |
| **Colonna** | **Tipo di dato** | **Attributi[[7]](#footnote-7)** |
| **Altezza** | DOUBLE | NN |
| **Categoria** | VARCHAR(45) | NN |
| **Codice** | VARCHAR(5) | PK,NN |
| **Descrizione** | VARCHAR(100) | NN |
| **Durata** | INT | NN |
| **InizioAsta** | DATETIME | NN |
| **Larghezza** | DOUBLE | NN |
| **Lunghezza** | DOUBLE | NN |
| **OffertaMassima** | DOUBLE |  |
| **PrezzoBase** | DOUBLE | NN |
| **Proprietario** | VARCHAR(30) |  |
| **Stato** | VARCHAR(45) | NN |
| **StatoAsta** | VARCHAR(20) | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella Cliente** | | |
| **Colonna** | **Tipo di dato** | **Attributi[[8]](#footnote-8)** |
| **Username cliente** | VARCHAR(30) | PK,NN |
| **Password cliente** | VARCHAR(45) | NN |
| **CAP** | VARCHAR(10) | NN |
| **CF** | VARCHAR(16) | NN |
| **CITTA’** | VARCHAR(50) | NN |
| **CITTA’ NASCITA** | VARCHAR(50) | NN |
| **CIVICO** | VARCHAR(5) | NN |
| **NOME** | VARCHAR(45) | NN |
| **COGNOME** | VARCHAR(45) | NN |
| **CVV** | VARCHAR(4) | NN |
| **DATA NASCITA** | DATE | NN |
| **DATA SCADENZA** | DATE | NN |
| **NUMERO CARTA** | VARCHAR(20) | UQ,NN |
| **USERNAME** | VARCHAR(30) | UQ,NN |
| **VIA** | VARCHAR(40) | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella Amministratori** | | |
| **Colonna** | **Tipo di dato** | **Attributi[[9]](#footnote-9)** |
| **Username admin** | VARCHAR(30) | PK,NN |
| **Password admin** | VARCHAR(45) | NN |

## Indici

Si omettono gli indici relativi alle chiavi primarie delle tabelle e quelli relativi alle foreign key relativi, in quanto questi vengono generati automaticamente; L’unico indice utilizzato è:

|  |  |
| --- | --- |
| **Tabella oggetto\_in\_asta** | |
| **Indice data\_fine\_asta** | **Tipo[[10]](#footnote-10):** |
| idx\_offerte\_appese | IDX |

Serve per la stored procedure elabora\_coda\_controfferte perché grazie ad esso ordina gli ordini pendenti per data, riducendo il numero di lettura che avvengono al minuto

## Trigger

Il trigger trg\_after\_insert\_controfferta di cui allegherò il codice sotto, si occupa di aggiungere le controfferte automatiche in una nuova tabella chiamata controfferte in coda, questo è un tassello fondamentale per il meccanismo di controfferta automatica

CREATE

DEFINER=`root`@`localhost`

TRIGGER `asteonline`.`trg\_after\_insert\_controfferta`

AFTER INSERT ON `asteonline`.`offerte`

FOR EACH ROW

BEGIN

DECLARE var\_offerta\_massima\_oggetto DOUBLE;

DECLARE var\_durata\_asta INT;

DECLARE var\_inizio\_asta\_oggetto DATETIME;

DECLARE var\_scadenza\_asta DATETIME;

DECLARE var\_vecchio\_offerente VARCHAR(30);

DECLARE var\_controfferta\_massima\_vecchia DOUBLE;

DECLARE var\_importo\_nuova\_controfferta DOUBLE;

SELECT

oia.offerta\_massima,

oia.durata,

oia.inizio\_asta

INTO

var\_offerta\_massima\_oggetto,

var\_durata\_asta,

var\_inizio\_asta\_oggetto

FROM

asteonline.oggetti\_in\_asta oia

WHERE

oia.codice = NEW.oggetto\_in\_vendita;

SET var\_scadenza\_asta = DATE\_ADD(var\_inizio\_asta\_oggetto, INTERVAL var\_durata\_asta DAY);

IF (asteonline.controllo\_data(var\_scadenza\_asta)) THEN

SELECT

o.cliente,

o.controfferta\_massima

INTO

var\_vecchio\_offerente,

var\_controfferta\_massima\_vecchia

FROM

asteonline.offerte o

WHERE

o.oggetto\_in\_vendita = NEW.oggetto\_in\_vendita

AND o.cliente != NEW.cliente

AND o.controfferta\_massima IS NOT NULL

AND o.controfferta\_massima > 0

ORDER BY

o.valore\_effettivo DESC,

o.controfferta\_massima DESC

LIMIT 1;

IF var\_vecchio\_offerente IS NOT NULL THEN

SET var\_importo\_nuova\_controfferta = NEW.valore\_effettivo + 0.50;

IF var\_importo\_nuova\_controfferta > var\_controfferta\_massima\_vecchia THEN

SET var\_importo\_nuova\_controfferta = var\_controfferta\_massima\_vecchia;

END IF;

IF (var\_importo\_nuova\_controfferta > NEW.valore\_effettivo) THEN

INSERT INTO asteonline.controfferte\_in\_coda (

var\_cliente,

var\_oggetto\_in\_vendita,

var\_offerta\_automatica,

var\_controfferta\_massima

)

VALUES (

var\_vecchio\_offerente,

NEW.oggetto\_in\_vendita,

var\_importo\_nuova\_controfferta,

var\_controfferta\_massima\_vecchia

);

END IF;

END IF;

END IF;

END

## Eventi

Ai fini della specifica si sono utilizzati 2 eventi :

-controllo\_chiusura\_aste : evento che si occupa ogni giorno di controllare eventuali aste scadute ed aggiornare lo stato delle aste (utilizzando la stored procedures chiudi\_asta)

CREATE EVENT `asteonline`.`controllo\_chiusura\_aste`

ON SCHEDULE EVERY 1 DAY

STARTS CURRENT\_TIMESTAMP

DO

BEGIN

    CALL asteonline.chiudi\_asta();

END //

DELIMITER ;

-elabora\_controfferte: evento che ogni minuto(per una migliore precisione può essere impostato con granularità minore) chiama la stored procedure elabora\_coda\_controfferte

CREATE EVENT `asteonline`.`elabora\_controfferte`

ON SCHEDULE EVERY 1 MINUTE

STARTS CURRENT\_TIMESTAMP

DO

BEGIN

    CALL asteonline.elabora\_coda\_controfferte();

END //

DELIMITER ;

Ovviamente per poter utilizzare questi eventi, mi sono dovuto accertare di avere on l’event scheduler con la seguente query sql: SET GLOBAL event\_scheduler = ON;

## Viste

Non sono state utilizzate viste.

## Stored Procedures e transazioni

1)CREATE DEFINER=`root`@`localhost` PROCEDURE `login`(

IN var\_username VARCHAR(30),

IN var\_password VARCHAR(80),

OUT var\_role VARCHAR(15)

)

BEGIN

DECLARE EXIT HANDLER FOR SQLEXCEPTION

BEGIN

ROLLBACK;

RESIGNAL;

END;

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;

START TRANSACTION

SET var\_role = NULL

IF asteonline.controllo\_username(var\_username, 'cliente') AND EXISTS (

SELECT 1 FROM cliente WHERE username = var\_username AND password\_cliente = MD5(var\_password)

) THEN

SET var\_role = 'cliente';

ELSEIF asteonline.controllo\_username(var\_username, 'amministratore') AND EXISTS (

SELECT 1 FROM amministratore WHERE username\_admin = var\_username AND password\_admin = MD5(var\_password)

) THEN

SET var\_role = 'amministratore';

ELSE

SIGNAL SQLSTATE '45009' SET MESSAGE\_TEXT = 'Credenziali non valide o utente non riconosciuto.';

END IF;

commit;

END

Questa è la stored procedure relativa al login

2) CREATE DEFINER=`root`@`localhost` PROCEDURE `registrazione`(

in var\_nome varchar(45),

in var\_cognome varchar(45),

in var\_CF varchar(16),

in var\_data\_nascita date,

in var\_città\_nascita varchar(50),

in var\_password varchar(80),

in var\_numero\_carta varchar(20),

in var\_CVV varchar(4),

in var\_data\_scadenza date,

in var\_via varchar(100),

in var\_città varchar(100),

in var\_CAP varchar(10),

in var\_username varchar(30),

in var\_civico varchar(5)

)

BEGIN

declare select\_role varchar(16);

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read uncommitted;

start transaction;

set select\_role = 'cliente' ;

if not(asteonline.controllo\_username(var\_username, select\_role)) then

signal sqlstate '45008' set message\_text = 'username già esistente.';

end if;

if not (asteonline.controllo\_CF(var\_CF)) then

signal sqlstate '45001' set message\_text = 'Il codice fiscale inserito non è valido.';

end if;

if not (asteonline.controllo\_carta(var\_numero\_carta)) then

signal sqlstate '45002' set message\_text = "Il numero della carta di credito non è valido.";

end if;

if not (asteonline.controllo\_CVV(var\_CVV)) then

signal sqlstate '45003' set message\_text = "Il codice CVV inserito non è valido.";

end if;

if not (asteonline.controllo\_CAP(var\_CAP)) then

signal sqlstate '45004' set message\_text = "Il CAP inserito non è valido.";

end if;

if not(asteonline.controllo\_data(var\_data\_scadenza)) then

signal sqlstate '45005' set message\_text = 'la carta di credito risulta scaduta.';

end if;

insert into cliente

(CAP,CF,città,città\_nascita,civico,cognome,

CVV,data\_nascita,data\_scadenza,nome,numero\_carta,

password\_cliente,username,via)

values (var\_CAP,var\_CF,var\_città,var\_città\_nascita,var\_civico,var\_cognome,

var\_CVV,var\_data\_nascita,var\_data\_scadenza,var\_nome,var\_numero\_carta,

md5(var\_password),var\_username,var\_via);

commit

END

Questa è la stored procedure relativa alla registrazione

3)CREATE DEFINER=`root`@`localhost` PROCEDURE `visualizza\_oggetti\_aggiudicati`(

var\_utente VARCHAR(30)

)

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read uncommitted;

start transaction;

SELECT descrizione,offerta\_massima as prezzo\_finale,

Altezza,Lunghezza,Larghezza,

stato\_asta

FROM oggetti\_in\_asta

where oggetti\_in\_asta.proprietario = var\_utente AND stato\_asta != 'aperta' ;

commit;

END

Questa è la stored procedure relativa alla visualizzazione degli oggetti aggiudicati dai singoli utenti

4) CREATE DEFINER=`root`@`localhost` PROCEDURE `visualizza\_aste\_aperte`()

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read uncommitted;

start transaction;

SELECT

codice,

descrizione,

prezzo\_base,

Altezza,

Lunghezza,

Larghezza,

stato,

categoria,

offerta\_massima,

stato\_asta,

CONCAT(

'Scadenza il ',

DATE\_FORMAT((inizio\_asta + INTERVAL durata DAY), '%d/%m'), #Formatta il giorno e mese di chiusura (es.01/01)

' alle ',

DATE\_FORMAT((inizio\_asta + INTERVAL durata DAY), '%H:%i'), #Formatta l'ora e i minuti di chiusura (es. 21:20)

' quindi mancano ',

#Calcola i giorni rimanenti

CASE

WHEN TIMESTAMPDIFF(DAY, NOW(), (inizio\_asta + INTERVAL durata DAY)) > 0

THEN CONCAT(TIMESTAMPDIFF(DAY, NOW(), (inizio\_asta + INTERVAL durata DAY)), ' giorni',

IF(TIMESTAMPDIFF(DAY, NOW(), (inizio\_asta + INTERVAL durata DAY)) > 1, '', ''), ', ')

ELSE ''

END,

#Calcola le ore rimanenti (mod 24 per non includere i giorni completi)

CASE

WHEN MOD(TIMESTAMPDIFF(HOUR, NOW(), (inizio\_asta + INTERVAL durata DAY)), 24) > 0

THEN CONCAT(MOD(TIMESTAMPDIFF(HOUR, NOW(), (inizio\_asta + INTERVAL durata DAY)), 24), ' ore',

IF(MOD(TIMESTAMPDIFF(HOUR, NOW(), (inizio\_asta + INTERVAL durata DAY)), 24) > 1, '', ''),

IF(MOD(TIMESTAMPDIFF(MINUTE, NOW(), (inizio\_asta + INTERVAL durata DAY)), 60) > 0, ' e ', '')) -- Aggiunge ' e ' solo se ci sono anche i minuti

ELSE ''

END,

#Calcola i minuti rimanenti (mod 60 per non includere ore complete)

CASE

WHEN MOD(TIMESTAMPDIFF(MINUTE, NOW(), (inizio\_asta + INTERVAL durata DAY)), 60) > 0

THEN CONCAT(MOD(TIMESTAMPDIFF(MINUTE, NOW(), (inizio\_asta + INTERVAL durata DAY)), 60), ' minuti',

IF(MOD(TIMESTAMPDIFF(MINUTE, NOW(), (inizio\_asta + INTERVAL durata DAY)), 60) > 1, '', ''))

ELSE ''

END

) AS scadenza\_e\_tempo\_residuo,

(select count(\*) from offerte where codice= oggetto\_in\_vendita) as numero\_offerte

FROM

oggetti\_in\_asta

WHERE

NOW() < (inizio\_asta + INTERVAL durata DAY) AND(stato\_asta = 'aperta');

commit;

END

Questa è la stored procedure relativa alle aste aperte

5) CREATE DEFINER=`root`@`localhost` PROCEDURE `visualizza\_asta\_in\_corso`(

IN var\_utente VARCHAR(30) -- Parametro per l'utente che ha fatto l'offerta

)

BEGIN

DECLARE EXIT HANDLER FOR SQLEXCEPTION

BEGIN

ROLLBACK;

RESIGNAL;

END;

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

START TRANSACTION;

SELECT

asta.codice,

asta.descrizione,

asta.prezzo\_base,

asta.inizio\_asta,

asta.altezza,

asta.lunghezza,

asta.larghezza,

asta.durata,

asta.stato,

asta.categoria,

asta.offerta\_massima,

CONCAT(

'Scadenza il ',

DATE\_FORMAT((asta.inizio\_asta + INTERVAL asta.durata DAY), '%d/%m'), #Formatta la data di scadenza (es. 01/01)

' alle ',

DATE\_FORMAT((asta.inizio\_asta + INTERVAL asta.durata DAY), '%H:%i'), #Formatta l'ora di scadenza (es. 21:20)

' quindi mancano ',

#Calcola e formatta i giorni rimanenti

CASE

WHEN TIMESTAMPDIFF(DAY, NOW(), (asta.inizio\_asta + INTERVAL asta.durata DAY)) > 0

THEN CONCAT(TIMESTAMPDIFF(DAY, NOW(), (asta.inizio\_asta + INTERVAL asta.durata DAY)), ' giorno')

END,

#Calcola e formatta le ore rimanenti (modulo 24 per escludere i giorni completi)

CASE

WHEN MOD(TIMESTAMPDIFF(HOUR, NOW(), (asta.inizio\_asta + INTERVAL asta.durata DAY)), 24) > 0

THEN CONCAT(MOD(TIMESTAMPDIFF(HOUR, NOW(), (asta.inizio\_asta + INTERVAL asta.durata DAY)), 24), ' ora')

END,

#Calcola e formatta i minuti rimanenti (modulo 60 per escludere le ore complete)

CASE

WHEN MOD(TIMESTAMPDIFF(MINUTE, NOW(), (asta.inizio\_asta + INTERVAL asta.durata DAY)), 60) > 0

THEN CONCAT(MOD(TIMESTAMPDIFF(MINUTE, NOW(), (asta.inizio\_asta + INTERVAL asta.durata DAY)), 60), ' minuto')

END

) AS scadenza\_e\_tempo\_residuo,

(SELECT COUNT(\*) FROM Offerte WHERE oggetto\_in\_vendita = asta.codice) AS NumeroOfferte

FROM

oggetti\_in\_asta AS asta

INNER JOIN

offerte AS user\_off ON asta.codice = user\_off.oggetto\_in\_vendita

AND user\_off.cliente = var\_utente

WHERE

(asta.inizio\_asta + INTERVAL asta.durata DAY) > NOW() AND (asta.stato\_asta = 'aperta')

GROUP BY

asta.codice, asta.descrizione, asta.inizio\_asta, asta.durata,

asta.prezzo\_base, asta.offerta\_massima, asta.stato

ORDER BY

asta.inizio\_asta;

COMMIT;

END

Questa è la stored procedure relativa alla visualizzazioni delle aste in corso (per asta in corso si intende l’asta dove un utente ha realizzato almeno un’offerta )

6) CREATE DEFINER=`root`@`localhost` PROCEDURE `registrazione\_amministratore`(

var\_username varchar(30),

var\_password varchar(45)

)

BEGIN

declare select\_role varchar(16);

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set select\_role = 'amministratore';

set transaction isolation level read uncommitted;

start transaction;

if (asteonline.controllo\_username(var\_username,select\_role)) then

signal sqlstate '45008' set message\_text = 'username già esistente.';

end if;

insert into amministratore(username\_admin,password\_admin) VALUES

(var\_username,md5(var\_password));

commit;

END  
Store Procedure relativa alla registrazione da parte degli utenti amministratori

7) CREATE DEFINER=`root`@`localhost` PROCEDURE `recupera\_categorie`()

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level repeatable read;

start transaction;

SELECT

nome\_categoria,

categoria\_superiore

FROM

categoria;

END

END

Store Procedure relativa al recupero di tutte le categorie presenti nel sistema

8) CREATE DEFINER=`root`@`localhost` PROCEDURE `fai\_offerta`(

in var\_username VARCHAR(30),

in var\_importo DOUBLE,

in var\_codice\_oggetto VARCHAR(5),

in var\_controfferta\_automatica DOUBLE

)

BEGIN

DECLARE var\_scadenza DATE;

DECLARE var\_inizio\_asta DATE;

DECLARE var\_durata INT;

DECLARE var\_prezzo\_base DOUBLE;

DECLARE var\_offerta\_massima\_attuale DOUBLE;

DECLARE var\_proprietario\_attuale VARCHAR(30);

DECLARE EXIT HANDLER FOR SQLEXCEPTION

BEGIN

ROLLBACK;

RESIGNAL;

END;

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

START TRANSACTION;

SELECT

prezzo\_base,

inizio\_asta,

durata,

offerta\_massima,

proprietario

INTO

var\_prezzo\_base,

var\_inizio\_asta,

var\_durata,

var\_offerta\_massima\_attuale,

var\_proprietario\_attuale

FROM

oggetti\_in\_asta

WHERE

codice = var\_codice\_oggetto;

-- Controllo granularità dell'importo (multiplo di 0.50)

IF (var\_importo \* 100) % 50 != 0 THEN

SIGNAL SQLSTATE '45110' SET MESSAGE\_TEXT = 'Granularità importo non rispettata. L''importo deve essere un multiplo di 0.50.';

END IF;

IF var\_inizio\_asta IS NULL THEN

SIGNAL SQLSTATE '45016' SET MESSAGE\_TEXT = 'Oggetto non trovato.';

END IF;

SET var\_scadenza = DATE\_ADD(var\_inizio\_asta, INTERVAL var\_durata DAY);

IF NOT (asteonline.controllo\_data(var\_scadenza)) THEN

SIGNAL SQLSTATE '45015' SET MESSAGE\_TEXT = 'Asta non valida o scaduta.';

END IF;

IF (asteonline.controllo\_carta(var\_username)) THEN

SIGNAL SQLSTATE '45005' SET MESSAGE\_TEXT = 'Carta di credito non valida per questo utente.';

END IF;

IF (var\_proprietario\_attuale = var\_username) THEN

SIGNAL SQLSTATE '45111' SET MESSAGE\_TEXT = 'Sei già il miglior offerente. Se vuoi aumentare, effettua una nuova offerta superiore.';

END IF;

-- Caso 1: È la prima offerta per questo oggetto (offerta\_massima\_attuale è NULL)

IF var\_offerta\_massima\_attuale IS NULL THEN

IF var\_importo >= var\_prezzo\_base THEN

UPDATE

oggetti\_in\_asta

SET

offerta\_massima = var\_importo,

proprietario = var\_username

WHERE

codice = var\_codice\_oggetto;

INSERT INTO offerte(

cliente,

oggetto\_in\_vendita,

valore\_effettivo,

controfferta\_massima)

VALUES(

var\_username,

var\_codice\_oggetto,

var\_importo,

var\_controfferta\_automatica);

ELSE

SIGNAL SQLSTATE '45100' SET MESSAGE\_TEXT = 'Importo non sufficiente. Deve essere maggiore del prezzo base.';

END IF;

-- Caso 2: Ci sono già offerte per l'oggetto (offerta\_massima\_attuale NON è NULL)

ELSE

IF (var\_importo > var\_offerta\_massima\_attuale) THEN

UPDATE oggetti\_in\_asta

SET

offerta\_massima = var\_importo,

proprietario = var\_username

WHERE

codice = var\_codice\_oggetto;

INSERT INTO offerte(

cliente,

oggetto\_in\_vendita,

valore\_effettivo,

controfferta\_massima)

VALUES(

var\_username,

var\_codice\_oggetto,

var\_importo,

var\_controfferta\_automatica);

ELSE

SIGNAL SQLSTATE '45100' SET MESSAGE\_TEXT = 'Importo non sufficiente. Devi superare l''offerta massima attuale.';

END IF;

END IF;

COMMIT;

END  
Store procedure principale, si occupa di far realizzare a tutti gli utenti le loro offerte

9) CREATE DEFINER=`root`@`localhost` PROCEDURE `elabora\_coda\_controfferte`()

BEGIN

DECLARE var\_id\_coda\_loop INT;

DECLARE var\_cliente\_coda\_loop VARCHAR(30);

DECLARE var\_oggetto\_coda\_loop VARCHAR(20);

DECLARE var\_offerta\_coda\_loop DOUBLE;

DECLARE var\_controfferta\_coda\_loop DOUBLE;

DECLARE var\_fine\_cursor INT DEFAULT FALSE;

DECLARE cur\_controfferte CURSOR FOR

SELECT var\_id\_coda, var\_cliente, var\_oggetto\_in\_vendita, var\_offerta\_automatica, var\_controfferta\_massima

FROM asteonline.controfferte\_in\_coda

WHERE var\_stato = 'pending'

ORDER BY var\_data\_richiesta ASC;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET var\_fine\_cursor = TRUE;

OPEN cur\_controfferte;

leggi\_coda: LOOP

FETCH cur\_controfferte INTO var\_id\_coda\_loop, var\_cliente\_coda\_loop, var\_oggetto\_coda\_loop, var\_offerta\_coda\_loop, var\_controfferta\_coda\_loop; #legge una riga alla volta

IF var\_fine\_cursor THEN

LEAVE leggi\_coda;

END IF;

START TRANSACTION;

BEGIN

DECLARE EXIT HANDLER FOR SQLEXCEPTION

BEGIN

ROLLBACK;

UPDATE asteonline.controfferte\_in\_coda

SET var\_stato = 'failed', var\_messaggio\_errore = 'Errore durante la chiamata a controfferta\_automatica'

WHERE var\_id\_coda = var\_id\_coda\_loop;

END;

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

START TRANSACTION;

CALL asteonline.controfferta\_automatica(

var\_cliente\_coda\_loop,

var\_oggetto\_coda\_loop,

var\_offerta\_coda\_loop,

var\_controfferta\_coda\_loop

);

UPDATE asteonline.controfferte\_in\_coda

SET var\_stato = 'processed', var\_messaggio\_errore = NULL

WHERE var\_id\_coda = var\_id\_coda\_loop;

COMMIT;

END;

END LOOP;

CLOSE cur\_controfferte;

END

END

Store procedure necessaria alla realizzazione del meccanismo di controfferta; nello specifico quando attivata da un evento ha il compito di processare le controofferte in attesa nella tabella.

10) CREATE DEFINER=`root`@`localhost` PROCEDURE `crea\_categoria\_senza\_macrocategoria`(

var\_nome\_categoria VARCHAR(45)

)

BEGIN

DECLARE v\_existing\_category\_name VARCHAR(45);

DECLARE v\_count\_macrocategoria INT; -- Anche questa deve essere dichiarata

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read uncommitted;

start transaction;

SELECT nome\_categoria INTO v\_existing\_category\_name

FROM categoria

WHERE nome\_categoria = var\_nome\_categoria;

IF v\_existing\_category\_name IS NOT NULL THEN

SIGNAL SQLSTATE '45001' SET MESSAGE\_TEXT = 'Errore: Categoria con questo nome esiste gia.';

END IF;

insert into categoria (nome\_categoria) values (var\_nome\_categoria);

commit;

END

La stored procedure qui sopra e quella successivamente hanno un compito simile: creare le categorie degli oggetti(senza macrocategoria sopra) e con macrocategoria sotto

11) CREATE DEFINER=`root`@`localhost` PROCEDURE `crea\_categoria\_con\_macrocategoria`(

var\_nome\_categoria varchar(45),

var\_nome\_macrocategoria varchar(45)

)

BEGIN

DECLARE v\_existing\_category\_name VARCHAR(45);

DECLARE v\_count\_macrocategoria INT; -- Anche questa deve essere dichiarata

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level repeatable read;

start transaction;

SELECT nome\_categoria INTO v\_existing\_category\_name

FROM categoria

WHERE nome\_categoria = var\_nome\_categoria;

IF v\_existing\_category\_name IS NOT NULL THEN

SIGNAL SQLSTATE '45001' SET MESSAGE\_TEXT = 'Errore: Categoria con questo nome esiste gia.';

END IF;

SELECT COUNT(\*)

INTO v\_count\_macrocategoria

FROM categoria

WHERE nome\_categoria = var\_nome\_macrocategoria;

insert into categoria(nome\_categoria,categoria\_superiore) values ( var\_nome\_categoria,var\_nome\_macrocategoria);

commit;

END

12) CREATE DEFINER=`root`@`localhost` PROCEDURE `controfferta\_automatica`(

IN var\_cliente\_param VARCHAR(30),

IN var\_oggetto\_in\_vendita\_param VARCHAR(20),

IN var\_offerta\_automatica\_param DOUBLE,

IN var\_controfferta\_massima\_param DOUBLE

)

BEGIN

DECLARE var\_offerta\_massima\_db DOUBLE;

DECLARE var\_controfferta\_massima\_locale DOUBLE;

DECLARE EXIT HANDLER FOR SQLEXCEPTION

BEGIN

RESIGNAL;

END;

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

START TRANSACTION;

SET var\_controfferta\_massima\_locale = var\_controfferta\_massima\_param;

SELECT offerta\_massima

INTO var\_offerta\_massima\_db

FROM oggetti\_in\_asta

WHERE codice = var\_oggetto\_in\_vendita\_param;

IF var\_offerta\_automatica\_param = var\_controfferta\_massima\_locale THEN

SET var\_controfferta\_massima\_locale = NULL;

END IF;

IF var\_offerta\_massima\_db <= var\_offerta\_automatica\_param THEN

INSERT INTO offerte(

cliente,

controfferta\_massima,

oggetto\_in\_vendita,

valore\_effettivo

)

VALUES(

var\_cliente\_param,

var\_controfferta\_massima\_locale,

var\_oggetto\_in\_vendita\_param,

var\_offerta\_automatica\_param

);

UPDATE oggetti\_in\_asta

SET offerta\_massima = var\_offerta\_automatica\_param,

proprietario = var\_cliente\_param

WHERE codice = var\_oggetto\_in\_vendita\_param

AND var\_offerta\_automatica\_param > offerta\_massima;

END IF;

COMMIT;

END  
Stored procedures necessaria al meccanismo di controfferta automatica; questa stored viene chiamata dalla stored procedures elabora\_coda\_controfferte

13) CREATE DEFINER=`root`@`localhost` PROCEDURE `chiudi\_asta`()

BEGIN

DECLARE v\_inizio\_asta DATE;

DECLARE v\_durata INT;

DECLARE v\_id\_oggetto INT;

DECLARE v\_fine\_asta DATE;

DECLARE v\_proprietario varchar(30);

DECLARE done INT DEFAULT FALSE;

DECLARE cur\_aste CURSOR FOR

select codice, inizio\_asta, durata,proprietario

from oggetti\_in\_asta

where stato\_asta = 'aperta';

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN cur\_aste;

read\_loop: LOOP

FETCH cur\_aste INTO v\_id\_oggetto, v\_inizio\_asta, v\_durata,v\_proprietario;

-- Uscire dal loop se il cursore è finito

if done then

leave read\_loop;

end if;

set v\_fine\_asta = DATE\_ADD(v\_inizio\_asta, INTERVAL v\_durata DAY);

if( v\_fine\_asta <= now()) then

if v\_proprietario is not null then

UPDATE oggetti\_in\_asta

SET stato\_asta = 'oggetto venduto'

WHERE codice = v\_id\_oggetto;

end if;

if v\_proprietario is null then

update oggetti\_in\_asta

set stato\_asta = 'oggetto non venduto'

where codice = v\_id\_oggetto;

end if;

end if;

END LOOP;

-- Chiudo il cursore

CLOSE cur\_aste;

commit;

END

Stored procedure che si occupa a seguito di una chiamata da parte di un evento di “chiudere le aste” modificando il loro stato.

13) CREATE DEFINER=`root`@`localhost` PROCEDURE `aggiungi\_oggetto\_in\_asta`(

in var\_inizio\_asta datetime,

in var\_durata int,

IN var\_descrizione VARCHAR(100),

IN var\_prezzo\_base DOUBLE UNSIGNED,

IN var\_altezza DOUBLE,

IN var\_lunghezza DOUBLE,

IN var\_larghezza DOUBLE,

IN var\_stato VARCHAR(45),

IN var\_categoria VARCHAR(45),

IN var\_proprietario VARCHAR(16),

IN var\_offerta\_massima DOUBLE,

IN var\_codice varchar(5)

)

BEGIN

DECLARE EXIT HANDLER FOR SQLEXCEPTION

BEGIN

ROLLBACK;

RESIGNAL;

END;

set transaction isolation level read uncommitted;

start transaction;

if not (asteonline.controllo\_data(var\_inizio\_asta)) then

signal sqlstate '45013' set message\_text = 'data inizio asta non valida!';

end if;

INSERT INTO oggetti\_in\_asta (

descrizione,

prezzo\_base,

altezza,

lunghezza,

larghezza,

stato,

categoria,

codice,

inizio\_asta,

durata,

stato\_asta

) VALUES (

var\_descrizione,

var\_prezzo\_base,

var\_altezza,

var\_lunghezza,

var\_larghezza,

var\_stato,

var\_categoria,

var\_codice,

var\_inizio\_asta,

var\_durata,

'aperta'

);

commit;

END

Store Procedures per gli amministratori, serve per aggiungere nuovi oggetti in asta.

1. Indicare con E le entità, con R le relazioni [↑](#footnote-ref-1)
2. Per “aste aperte” si intende tutte le aste attive nel sistema [↑](#footnote-ref-2)
3. Per “aste in corso” si intende le aste in cui il singolo utente ha fatto almeno un’offerta. [↑](#footnote-ref-3)
4. [↑](#footnote-ref-4)
5. PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna. [↑](#footnote-ref-5)
6. [↑](#footnote-ref-6)
7. [↑](#footnote-ref-7)
8. PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna. [↑](#footnote-ref-8)
9. [↑](#footnote-ref-9)
10. IDX = index, UQ = unique, FT = full text, PR = primary. [↑](#footnote-ref-10)