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

**Basi di Dati**

**Progetto A.A. 2022/2023**

Sistema di Gestione di Aste Online

0267355

Alessandro Cortese

**Indice**

1. Descrizione del Minimondo 2

2. Analisi dei Requisiti 4

3. Progettazione concettuale 8

4. Progettazione logica 14

5. Progettazione fisica 27

# Descrizione del Minimondo

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  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  42  43  44 | 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 import 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** |
| 4 | Oggetto | Oggetto in asta | Un utente può effettuare offerte su un oggetto su cui è ancora aperta un’asta, non su un qualsiasi oggetto. |
| 15 | Asta | Oggetto | Una categoria viene assegnata ad un oggetto, non ad un’asta. |
| 39 | Transazioni | Offerte | La funzionalità “controfferta automatica” genera offerte, non transazioni. |

### 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. Gli utenti del sistema, previa registrazione, hanno la possibilità di fare offerte su un qualsiasi oggetto in asta. Al termine dell’asta, l’offerta maggiore sarà quella che avrà vinto l’asta. Alla registrazione, gli utenti devono comunicare il codice fiscale, il nome, il cognome, la data di nascita, la città di nascita, le informazioni sulla propria carta di credito (intestatario, numero, data di scadenza, codice CVV). Inoltre, essi devono fornire un indirizzo cui consegnare eventuali oggetti acquistati.  Gli amministratori gestiscono l’inserimento degli oggetti. Ogni oggetto è caratterizzato da un codice alfanumerico univoco, da una descrizione, da uno stato (ad esempio “come nuovo”, “in buone condizioni”, “non funzionante”, ecc.), da un prezzo di base d’asta, e da una descrizione delle dimensioni. Quando viene inserito un nuovo oggetto nel sistema, gli amministratori possono decidere la durata dell’asta, da un minimo di un giorno ad un massimo di sette giorni. Inoltre, a 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.  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 import 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.  Gli utenti, in ogni momento, possono visualizzare l’elenco degli oggetti aggiudicati e l’elenco degli oggetti per i quali è presente un’asta in corso cui hanno fatto almeno un’offerta. |

## Glossario dei Termini

|  |  |  |  |
| --- | --- | --- | --- |
| **Termine** | **Descrizione** | **Sinonimi** | **Collegamenti** |
| Utente | Persone che possono registrarsi all’interno della piattaforma, possono fare offerte su degli oggetti in asta e aggiudicarsi un’asta per acquistare un oggetto. |  | Offerte, Oggetto in Asta |
| Oggetto in asta | Oggetto su cui più utenti possono fare delle offerte. Ad ogni oggetto viene associata una categoria. |  | Utente, Offerta, Categoria |
| Offerte | Offerta, con un importo, relativo ad un oggetto in asta effettuata da un utente. |  | Utente, Oggetto in Asta |
| Categoria | Classificazione degli oggetti inseriti nel sistema, organizzate in un titolario gerarchico organizzato su tre livelli. |  | Oggetto in Asta |

## Raggruppamento dei requisiti in insiemi omogenei

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

|  |
| --- |
| **Frasi relative a Oggetto in Asta** |
| 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.  A ciascun oggetto viene associato una categoria.  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.  Gli utenti, in ogni momento, possono visualizzare l’elenco degli oggetti aggiudicati. |

|  |
| --- |
| **Frasi relative a Offerta** |
| Gli utenti del sistema hanno la possibilità di fare offerte su un qualsiasi oggetto in asta. Al termine dell’asta, l’offerta maggiore sarà quella che avrà vinto l’asta.  Dato un oggetto in asta, gli utenti possono fare un’offerta, maggiore del valore attuale di offerta.  La granularità di incremento delle offerte è di multipli di 50 centesimi di euro. Inoltre, un utente che ha attualmente piazzato l’offerta massima, può sfruttare la funzionalità di “controfferta automatica”. Tale funzionalità permette all’utente di indicare un importo massimo con cui si intende rilanciare l’offerta, qualora un altro utente faccia un’offerta maggiore. La gestione delle offerte pertanto funziona nel modo seguente. L’utente A indica un importo I con cui vuole rilanciare l’offerta nei confronti dell’utente B che è attualmente il migliore offerente. L’utente B ha anche indicato un importo di controfferta C. Se C > I, il sistema indicherà come miglior offerente l’utente A, con importo temporaneo I, ma immediatamente dopo indicherà nuovamente l’utente B come migliore offerente, con un import 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.  Gli utenti, in ogni momento, possono visualizzare l’elenco degli oggetti per i quali è presente un’asta in corso cui hanno fatto almeno un’offerta. |

|  |
| --- |
| **Frasi relative a Categoria** |
| Le categorie appartengono ad un titolario gerarchico, organizzato su un massimo di tre livelli. La gestione delle categorie degli oggetti afferisce sempre agli amministratori del sistema. |

# Progettazione concettuale

## Costruzione dello schema E-R

Per la costruzione del modello concettuale si utilizza una strategia mista. Nella Figura 1 viene raffigurato lo schema scheletro costruito a partire dalle specifiche, rappresentate a livello astratto i concetti principali dell’applicazione.

### Immagine che contiene diagramma Descrizione generata automaticamente

Lo schema scheletro rappresentato cattura vari concetti:

* L’Utente, che è rilevante ai fini dell’applicazione, viene rappresentato con un’entità;
* L’Oggetto in Asta è un altro concetto rilevante per il sistema, anche lui viene rappresentato come un’entità;
* L’Offerta rappresenta il concetto di offerta che un Utente può fare su un particolare Oggetto in Asta, viene pertanto rappresentato come un’entità;
* La Categoria cattura il concetto di categorie in cui possono essere divisi i vari oggetti in asta, viene rappresentato come un’entità;
* Effettuare cattura la possibilità di un Utente di fare una particolare offerta su un preciso Oggetto in Asta, viene rappresentato come un’associazione;
* Riferimento associa Offerta con Oggetto in Asta, associazione che cattura il concetto rappresentante una precisa offerta fa riferimento ad un particolare oggetto in asta;
* Assegnato associa Oggetto in Asta con Categorie, indica che un Oggetto in Asta ha associata una Categoria, viene rappresentato con un’associazione.

In riferimento ai concetti introdotti, si utilizzano dei raffinamenti successivi per ottenere una completa rappresentazione del mini-mondo di riferimento.

In Figura 2 viene raffigurata l’entità Utente, rappresentate un utente del sistema in grado di fare offerte per poter acquistare un particolare oggetto in asta.

Immagine che contiene diagramma

Descrizione generata automaticamente

L’entità è indentificata del codice fiscale, gli altri attributi caratterizzante l’entità sono: il nome, il cognome, la data di nascita, la città di nascita, la carta di credito (con le informazioni sulla data di scadenza, il codice CVV e il numero della carta stessa) e l’indirizzo di consegna al quale consegnare eventuali oggetti acquistati (con informazioni relative alla via, numero civico, comune e CAP).

In Figura 3 viene raffigurato il concetto relativo ad un Oggetto in Asta.

Immagine che contiene testo, diagramma, linea, schermata

Descrizione generata automaticamente

Gli Oggetti in Asta vengono identificati da un codice alfanumerico univoco, gli altri attributi sono: una descrizione, uno stato (ad esempio “come nuovo”, “in buone condizioni”, “non funzionante”, ecc.), da una descrizione delle dimensioni, da un prezzo di base d’asta, dalla durata dell’asta sullo specifico oggetto e dall’istante di inizio dell’asta sull’oggetto (con le informazioni relative alla data e all’orario).

Viene utilizzato il costrutto della generalizzazione per catturare il concetto di un Oggetto Venduto acquistato da un Utente dell’applicazione, introducendo l’attributo prezzo di vendita rappresentate il prezzo con cui viene venduto l’oggetto considerato, e il concetto di Oggetto non Venduto rappresentate un oggetto messo in asta a cui non vengono fatte offerte entro il tempo limite della fine dell’asta.

In Figura 4 troviamo la raffinazione dell’entità Offerta e delle relazioni ricavate finora.

Immagine che contiene diagramma, Piano, Disegno tecnico, linea

Descrizione generata automaticamente

L’entità offerta cattura il concetto di offerta fatta da un Utente su un preciso Oggetto in Asta. Dato che un Utente nel corso dell’asta per quell’oggetto può fare più di una offerta per acquistarlo, si è deciso di identificare l’entità con un identificatore esterno verso le entità Utente e Oggetto in Asta con l’aggiunta dell’attributo Importo, utilizzato per rappresentare il valore della specifica offerta.

Viene utilizzato il costrutto della generalizzazione per catturare il concetto di Controfferta Automatica, introducendo l’attributo di importo massimo.

L’entità Offerta è caratterizzata anche dall’attributo istante, utilizzato per catturare la data e l’orario di quando è stata fatta una particolare Offerta su un particolare Oggetto in Asta.

Le associazioni che abbiamo rappresentato finora dalla specifica sono:

* L’associazione Effettuare relativo al concetto che un utente può fare più offerte, le cardinalità sono (0, N) da parte di Utente e (1, 1) da parte di Offerta, quest’ultima cardinalità consente di utilizzare l’identificatore esterno di Offerta verso Utente;
* L’associazione Riferimento fra l’entità Offerta e Oggetto in Asta, tale associazione indica che una particolare Offerta è riferita ad un Oggetto, le cardinalità sono (1, 1) da parte di Offerta e (0, N) da parte di Oggetto in Asta, dove la cardinalità (1, 1) di Offerta consente di utilizzare l’identificatore esterno per l’entità Offerta;
* L’associazione Possiede tra le entità Utente e l’entità Oggetto Venduto, ciò a rappresentare che un Utente può acquistare e quindi diventare il possessore di un Oggetto in Asta Venduto. Le cardinalità sono (0, N) da parte dell’Utente e (1, 1) da parte di Oggetto Venduto, questo perché un Utente registrato nel sistema può non aver acquistato ancora nessun oggetto oppure averne acquistato più di uno, mentre un Oggetto Venduto viene acquistato da un solo Utente.

In Figura 5 troviamo il raffinamento del concetto di Categoria, del titolario gerarchico in cui vengono organizzate le stesse all’interno del sistema, e dell’associazione tra l’Oggetto in Asta e il concetto di Categoria.

Immagine che contiene diagramma, schizzo, Piano, Disegno tecnico

Descrizione generata automaticamente

L’entità Categoria viene identificata dal nome. Utilizziamo la relazione ricorsiva sull’entità Categoria per rappresentare il titolario gerarchico, dove una Categoria può avere più sottocategorie. Come da specifica, un Oggetto in Asta viene associato ad una Categoria, concetto rappresentato dalla relazione Assegnato.

### Integrazione finale

Immagine che contiene diagramma, Piano, schizzo, Disegno tecnico

Descrizione generata automaticamente

## Regole aziendali

## L’Utente NON DEVE poter fare una nuova offerta su un oggetto che non è in asta;

* Un utente con la carta di credito scaduta NON DEVE poter acquistare un oggetto;
* La granularità dell’importo delle offerte DEVE essere di 0,50€;
* Il valore dell’importo della prima offerta DEVE essere almeno pari al prezzo di base d’asta per un oggetto in asta;
* Il valore dell’importo di una nuova offerta su un oggetto in asta su cui sono state già fatte delle offerte DEVE essere maggiore del valore dell’importo di tutte le offerte precedenti;
* Il valore del prezzo di vendita di un oggetto in asta SI OTTIENE dal valore dell’importo dell’ultima offerta su quell’oggetto;
* Un utente che acquista un oggetto DEVE aver fatto l’offerta vincente su quell’oggetto in asta;
* Il numero di offerte fatte su un oggetto SI OTTIENE contando il numero di offerte fatte su quell’oggetto;
* La durata dell’asta DEVE essere compresa tra 1 e 7 giorni;
* Il tempo mancante per la chiusura dell’asta SI OTTIENE dalla data di terminazione dell’asta e dalla data odierna;
* La data di terminazione dell’asta SI OTTIENE sommando la data di inizio dell’asta con la durata dell’asta;
* Il livello gerarchico massimo delle categorie DEVE essere 3.

## Dizionario dei dati

|  |  |  |  |
| --- | --- | --- | --- |
| **Entità** | **Descrizione** | **Attributi** | **Identificatori** |
| Utente | Colui che utilizza il sistema | Nome, Cognome, Data di Nascita,  Città di Nascita,  Carta di Credito, Indirizzo di Consegna. | CF |
| Oggetto in Asta | Oggetto che viene messo in asta | Descrizione, Stato, Prezzo di base d’asta, Descrizione Dimensioni, Durata asta, Inizio asta | Codice |
| Oggetto Venduto | Oggetto in asta venduto ad un utente | Prezzo di vendita | Codice |
| Oggetto non Venduto | Oggetto in asta che non viene venduto |  | Codice |
| Offerta | Offerta fatte da un Utente su un particolare Oggetto in Asta | Istante | Importo, CF (Utente), Codice (Oggetto in Asta) |
| Controfferta Automatica | Controfferta automatica su un particolare Oggetto in Asta | Importo Massimo | Importo, CF (Utente), Codice (Oggetto in Asta) |
| Categoria | Categoria assegnata ai vari Oggetti in Asta |  | Nome |

|  |  |  |  |
| --- | --- | --- | --- |
| **Relazione** | **Descrizione** | **Entità Coinvolte** | **Attributi** |
| Effettuare | Associa Utente ad Offerta | Utente (0, N), Offerta (1, N) |  |
| Possiede | Associa un Utente all’Oggetto Venduto che acquista | Utente (0, N),  Oggetto Venduto (1, 1) |  |
| Riferimento | 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

* 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.
* Si suppone che il numero di Utenti che utilizzino il sistema è pari a 1.500.
* Ogni utente, in media, partecipa all’8% delle aste per un oggetto.
* Un utente fa dalle 2 alle 6, in media 4, offerte per un oggetto in asta.
* In media il 2% degli utenti che partecipano all’asta utilizza la funzionalità “controfferta automatica”, che in media genera 15 offerte.
* Il numero degli oggetti in asta è pari a 1.000.
* Il numero di oggetti venduti è di 700.
* Il numero di oggetti non venduti è di 300.
* Il numero di categorie è 30.

|  |  |  |
| --- | --- | --- |
| **Concetto nello schema** | **Tipo[[1]](#footnote-1)** | **Volume atteso** |
| Utente | E | 1.500 |
| Oggetto in Asta | E | 1.000 |
| Oggetto Venduto | E | 700 |
| Oggetto non Venduto | E | 300 |
| Offerta | E | 354.000 |
| Controfferta Automatica | E | 32.000 |
| Categoria | E | 30 |
| Effettuare | R | 386.000 |
| Riferimento | R | 386.000 |
| Possiede | R | 700 |
| Assegnato | R | 1.000 |
| Gerarchia | R | 30 |

## Tavola delle operazioni

|  |  |  |
| --- | --- | --- |
| **Cod.** | **Descrizione** | **Frequenza attesa** |
| L1 | Login | 20/settimana |
| L2 | Registrazione | 25/mese |
| U1 | Fare un’offerta | 10/settimana |
| U2 | Fare un’offerta impostando la controfferta automatica | 2/settimana |
| U3 | Visualizza aste aperte | 20/mese |
| U4 | Visualizza Oggetti Aggiudicati | 7/mese |
| U5 | Visualizza aste aperte con almeno un’offerta | 15/settimana |
| AM1 | Inserimento Oggetto in Asta | 34/mese |
| AM2 | Inserimento di una nuova Categoria | 1/anno |
| AM3 | Modifica di una Categoria | 1/anno |
| AM4 | Cancellazione di una Categoria | 1/anno |
| OP1 | Spostamento dei dati più vecchi di cinque anni | 1/ogni 5 anni |

## Costo delle operazioni

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

*Tabella 1: Operazione L1- Login*

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Utente | E | 1 | L |

Il costo dell’operazione è di 1 accessi, considerando la frequenza dell’operazione otteniamo 20 accessi alla settimana.

*Tabella 2: Operazione L2 – Registrazione*

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Utente | E | 1 | S |

Il costo dell’operazione è di 2 accessi, considerando la frequenza dell’operazione otteniamo 30 accessi al mese.

*Tabella 3: Operazione U1 – Fare un’offerta*

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Utente | E | 1 | L |
| Oggetto in Asta | E | 1 | L |
| Riferimento | R | 386 | L |
| Offerta | E | 354 | L |
| Controfferta Automatica | E | 32 | L |
| Effettuare | R | 1 | S |
| Offerta | E | 1 | S |
| Riferimento | R | 1 | S |

Per effettuare l’operazione bisognerà prima verificare che il valore dell’importo della nuova offerta sia maggiore di tutti di importi di tutte le precedenti offerte. Pertanto, per effettuare il controllo dato l’Oggetto in Asta su cui l’Utente vuole fare un’offerta, bisognerà controllare l’importo di tutte le offerte fatte per l’oggetto, si suppone che le offerte per un Oggetto in Asta siano in media 386, contado anche le offerte di Controfferta Automatica. Solo dopo aver fatto il controllo, si può procedere con la scrittura dell’offerta per lo specifico Oggetto in Asta.

Il costo dell’operazione è di 780 accessi, che diventano 7.800 accessi settimanali considerando la frequenza dell’operazione.

*Tabella 4: Operazione U2 – Fare un’offerta impostando la controfferta automatica*

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Utente | E | 1 | L |
| Oggetto in Asta | E | 1 | L |
| Riferimento | R | 386 | L |
| Offerta | E | 354 | L |
| Controfferta Automatica | E | 32 | L |
| Effettuare | R | 1 | S |
| Controfferta Automatica | E | 1 | S |
| Riferimento | R | 1 | S |

Nel caso in cui l’offerta venga generata impostando la funzionalità di controfferta automatica, l’unica cosa differente dal costo analizzato poco sopra, è che la scrittura finale dell’offerta avverrà sull’entità Controfferta Automatica e non su Offerta impostando il valore dell’importo massimo. Il sistema poi genererà il resto delle offerte per conto di quegli utenti che hanno attivato tale funzionalità.

Il costo dell’operazione è di 780 accessi, che considerando la frequenza delle operazioni diventano 1.560 accessi settimanali.

*Tabella 5: Operazione U3 – Visualizza aste aperte*

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Oggetto in Asta | E | 17 | L |
| Offerta | E | 6.018 | L |
| Controfferta Automatica | E | 544 | L |
| Riferimento | R | 6.562 | L |

L’operazione consente ad un utente di visualizzare lo stato delle aste aperte, che supponiamo essere 17 al mese, e quindi di sapere anche il numero di offerte fatte, sia le offerte generate dagli utenti manualmente che tramite la funzionalità di controfferta automatica, l’importo dell’offerta maggiore e il tempo mancante alla fine dell’asta. La relazione Riferimento, l’entità Offerta e l’entità Controfferta Automatica, vengono accedute per contare tutte le offerte per un preciso oggetto in asta e per cercare l’importo dell’offerte migliore, attuale, per un particolare oggetto in asta.

Il costo dell’operazione è di 13.141 accessi, che diventano 262.820 accessi al mese considerando la frequenza.

*Tabella 6: Operazione U4 – Visualizza Oggetti Aggiudicati*

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Utente | E | 1 | L |
| Possiede | R | 1 | L |
| Oggetto Venduto | E | 1 | L |

Per calcolare il costo dell’operazione si sono utilizzati numeri interi al posto dei numeri decimali, questo perché con l’assunzione dei volumi fatta in precedenza un Utente non possiede in media un Oggetto Venduto, ma bensì 0,466.

Con l’assunzione fatta, il costo dell’operazione ammonta a 3 accessi, che, considerando la frequenza, diventano 21 accessi mensili.

*Tabella 7: Operazione U5 – Visualizza Aste Aperte con almeno un’offerta*

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Utente | E | 1 | L |
| Effettuare | R | 772 | L |
| Offerta | E | 708 | L |
| Controfferta Automatica | E | 64 | L |
| Riferimento | R | 772 | L |
| Oggetto in Asta | E | 2 | L |

Per la stima del costo dell’operazione si assume che un utente partecipi in media a due aste, rispettivamente per due oggetti in asta. Supponendo che il numero di offerte per un oggetto in asta sia di 354 relative alle offerte generate manualmente che 54 per le offerte generate dalla funzionalità Controfferta Automatica, per i due oggetti che segue allora bisognerà considerare l’accesso in lettura per le 772 offerte totali per i due oggetti.

Il costo dell’operazione ammonta ad un totale di 2.319 accessi, che diventano 34.785 accessi alla settimana considerando la frequenza dell’operazione.

*Tabella 8: Operazione AM1 – Inserimento Oggetto in Asta*

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Oggetto in Asta | E | 1 | S |

Il costo dell’operazione è di 2 accessi, che considerando la frequenza dell’operazione diventano 68 accessi al mese.

*Tabella 9: Operazione AM2 – Inserimento di una nuova Categoria*

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Categoria | E | 3 | S |
| Gerarchia | R | 3 | L |
| Gerarchia | R | 1 | S |

Il costo dell’operazione consiste in 8 accessi. Le letture preventiva della relazione Gerarchia è giustificata del fatto che per ogni categoria il livello massimo di profondità deve essere 3, quindi prima di inserire una nuova categoria bisogna controllare che tale vincolo non venga infranto. Considerando la frequenza, otteniamo 8 accessi l’anno.

*Tabella 10: Operazione AM3 – Modifica di una Categoria*

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Categoria | E | 1 | L |
| Gerarchia | R | 1 | L |
| Categoria | E | 1 | S |
| Gerarchia | R | 1 | S |

Nell’operazione relativa alla modifica di una categoria bisognerà prima recuperare la categoria da modificare e il relativo livello gerarchico, solo a questo punto possiamo modificare i dati inseriti precedentemente.

Il costo ammonta a 6 accessi. Considerando la frequenza dell’operazione otteniamo 6 accessi l’anno.

*Tabella 11: Operazione AM4 – Cancellazione di una Categoria*

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Categoria | E | 1 | L |
| Gerarchia | R | 1 | L |
| Categoria | E | 1 | S |
| Gerarchia | R | 1 | S |

Il costo dell’eliminazione di una categoria ha le stesse supposizioni della modifica di una categoria.

Il costo dell’operazione consiste in 6 accessi, che, considerando la frequenza dell’operazione, diventano 6 accessi annuali.

*Tabella 12: Operazione OP1 – Spostamento dei dati più vecchi di cinque anni*

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Oggetto in Asta | E | 1.000 | L |
| Oggetto Venduto | E | 700 | L |
| Oggetto non Venduto | E | 300 | L |
| Offerta | E | 354.000 | L |
| Controfferta Automatica | E | 32.000 | L |
| Effettuare | R | 386.000 | L |
| Riferimento | R | 386.000 | L |
| Possiede | R | 700 | L |
| Assegnato | R | 1.000 | L |
| Oggetto in Asta | E | 1.000 | S |
| Oggetto Venduto | E | 700 | S |
| Oggetto non Venduto | E | 300 | S |
| Offerta | E | 354.000 | S |
| Controfferta Automatica | E | 32.000 | S |
| Effettuare | R | 386.000 | S |
| Riferimento | R | 386.000 | S |
| Possiede | R | 700 | S |
| Assegnato | R | 1.000 | S |

Il costo dell’operazione prevede la lettura dei dati inseriti in cinque anni di utilizzo del sistema, quindi la lettura di tutti e lo spostamento, e quindi una scrittura, degli stessi in un’altra base di dati.

Il costo ammonta è di 3.485.100 accessi. Il costo dell’operazione, considerando la frequenza, è di 3.485.100 accessi ogni 5 anni.

## 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 \* 1.000 Oggetto in Asta = 4.000 B = 4 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 \* 1.000 Oggetto in Asta = 8.000 B = 8 kB

Di memoria utilizzata.

Considerando entrambe le ridondanze abbiamo un totale di 12 kB di memoria utilizzata nella base di dati.

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

*Tabella 13: Operazione U1 - Fare un’offerta*

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Utente | E | 1 | L |
| Oggetto in Asta | E | 1 | L |
| Riferimento | R | 386 | L |
| Offerta | E | 354 | L |
| Controfferta Automatica | E | 32 | L |
| Effettuare | R | 1 | S |
| Offerta | E | 1 | S |
| Riferimento | R | 1 | S |
| Oggetto in Asta | E | 1 | S |

*Tabella 14: Operazione U2 - Fare un’offerta impostando la controfferta automatica*

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Utente | E | 1 | L |
| Oggetto in Asta | E | 1 | L |
| Riferimento | R | 386 | L |
| Offerta | E | 354 | L |
| Controfferta Automatica | E | 32 | L |
| Effettuare | R | 1 | S |
| Controfferta Automatica | E | 1 | S |
| Riferimento | R | 1 | S |
| Oggetto in Asta | E | 1 | S |

In entrambe le operazioni la scrittura in più dell’entità Oggetto in Asta è relativa all’aggiornamento del numero di offerte relative a quell’oggetto e per aggiornare il valore relativo al mantenimento del valore della massima offerta; quindi, in presenza della ridondanza, paghiamo 2 accessi in più in confronto al costo delle stesse operazioni senza ridondanza, con un costo totale di 782 accessi per entrambe le operazioni

*Tabella 15: Operazione U3 - Visualizza aste aperte– In presenza delle ridondane*

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Oggetto in Asta | E | 17 | L |

Considerando la frequenza con cui vengono eseguite le operazioni:

* Operazione U1 senza ridondanza: 7.800 accessi/settimana;
* Operazione U1 con ridondanza: 7.820 accessi/settimana;
* Operazione U2 senza ridondanza: 1.560 accessi/settimana;
* Operazione U2 con ridondanza: 1564 accessi/settimana;
* Operazione U3 senza ridondanza: 262.820 accessi/mese che corrispondono a circa 65.705 accessi/settimana;
* Operazione U3 con ridondanza: 17 accessi/mese che corrispondono a 4,25, utilizzando numeri interi, 5 accessi/settimana.

In conclusione, INTRODUCIAMO la ridondanza relativa al numero di offerte e per il valore della massima offerta per un Oggetto in Asta; quindi, utilizziamo 12 kB di memoria in più. Tale scelta viene presa al fine di non sovraccaricare troppo l’esecuzione dell’operazione U3 perché in questo modo per reperire il numero di offerte fatte per un oggetto in asta basterà solo leggere il dato dall’entità Oggetto in Asta e non reperirlo tramite il conteggio, la stessa cosa vale per reperire il valore della massima offerta per un Oggetto in Asta, cioè leggerlo direttamente dall’entità piuttosto che reperirlo in base all’ultima offerta effettuata sull’oggetto. La scelta è giustificata anche dal fatto che l’operazione U3 ha un costo in termini di accessi superiore alle operazioni U1 ed U2.

*Fine Asta*

Il dato relativo alla fine dell’asta per un oggetto rappresenta una ridondanza per l’entità Oggetto in Asta poiché può essere ottenuto sommando la data di inizio dell’asta con la durata della stessa. La deduzione di questo dato viene eseguito anche nell’operazione U3, che è una delle due operazioni che in qualche modo coinvolgono la ridondanza, che viene eseguita 20 volte al mese e che quindi tutte le volte dovrà ottenere la data di fine dell’asta. L’altra operazione che coinvolge la ridondanza è l’operazione AM1 relativa all’inserimento di un Oggetto in asta nel sistema, che rappresenta una scrittura nella base di dati.

Supponendo che la data di terminazione dell’asta venga mantenuta con un tipo di dato stringa formata da 8 caratteri, sono necessari 8 B per il mantenimento di una data. Considerando la tavola dei volumi riportata sopra, otteniamo:

8 B \* 1.000 = 8.000 B = 8 kB

di memoria utilizzata.

Al fine di non appesantire troppo l’esecuzione dell’operazione U3, si decide di INTRODURRE la ridondanza. Un altro motivo dell’introduzione di tale ridondanza verrà discusso più avanti quando verranno discussi gli indici utilizzati nella progettazione fisica della base di dati, sezione 5 – Progettazione Fisica.

In Figura 7 viene raffigurato lo schema E-R con l’aggiunta delle due ridondanze

**Immagine che contiene diagramma, Piano, Disegno tecnico, schizzo

Descrizione generata automaticamente**

**Eliminazione delle Generalizzazioni**

*Oggetto in Asta e Figlie*

Per quanto riguarda l’entità Oggetto in Asta e le sue generalizzazioni Oggetto Venduto e Oggetto non Venduto potevamo optare per le tre classiche strategie (accorpamento delle entità figlie della generalizzazione nel genitore, accorpamento del genitore della generalizzazione nelle figlie e sostituzione della generalizzazione con associazioni) oppure usare una soluzione mista.

In base alle operazioni e agli accessi, scegliamo di accorpare le entità figlie nel genitore aggiungendo un attributo in grado di distinguere l’occorrenza delle entità, viene aggiunto anche l’attributo relativo al Prezzo di Vendita, presente inizialmente nell’entità Oggetto Venduto, nell’entità genitore Oggetto in Asta però questa volta sarà un attributo opzionale, poiché un oggetto non venduto non avrà nessun prezzo di vendita.

Eliminando la generalizzazione in questo modo dovremmo modificare la relazione che c’era precedentemente tra Utente e Oggetto Venduto; avendo accorpato le entità figlie al padre l’associazione Possiede sarà tra le entità Utente e Oggetto in Asta, dove le cardinalità da parte di Utente rimangono invariate e la cardinalità da parte di Oggetto in Asta diventa opzionale, cioè avrà come cardinalità della relazione (0, 1).

*Offerta e Controfferta Automatica*

Come nel caso di Oggetto in Asta e figlie, possiamo applicare una delle soluzioni sopra elencate. Anche io questo caso, si è scelto di accorpare l’entità figlia all’entità padre, cioè accorpare l’entità Controfferta Automatica nell’entità Offerta. Aggiungiamo un attributo in grado di distinguere il tipo di occorrenza di entità e l’attributo relativo all’importo massimo, che però sarà opzionale quindi con cardinalità (0, 1).

## Trasformazione di attributi e identificatori

Per poter rappresentare ogni attribuito nel modello relazionale bisogna prima eliminare tutti gli attributi composti, basterà eliminare gli attributi composti ed aggiungere direttamente i singoli attributi componenti.

In Figura 8 viene rappresentato lo schema E-R ristrutturato.

Immagine che contiene diagramma, origami, schizzo, Piano

Descrizione generata automaticamente

## Traduzione di entità e associazioni

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

* *Utente*(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, Comune di Consegna, CAP di Consegna)
* *Oggetto in Asta*(Codice, Descrizione, Stato, Prezzo di Base, Descrizione Dimensioni, Tipo, Numero Offerte\*, Valore Massima Offerta\*, Data Inizio Asta, Durata Asta, Data Fine Asta, Orario Inizio Asta, Categoria, Prezzo di Vendita\*, Utente\*)
* *Oggetto in Asta*(Categoria) ⊆ *Categoria*(Nome)
* *Oggetto in Asta*(Utente) ⊆  *Utente*(CF)
* *Offerta*(Utente, Importo, Oggetto In Asta, Data, Orario, Tipo, Importo Massimo\*)
* *Offerta*(Utente) ⊆ *Utente*(CF)
* *Offerta*(Oggetto In Asta)⊆ *Oggetto In Asta*(Codice)
* *Categoria*(Nome, Macrocategoria\*)
* *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 U1, U2, U3, U4, U5.
* Amministratori:
  + Grant in esecuzione sulle operazioni AM1, AM2, AM3, AM4.

Si osservi che, per far si 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 amministratori** | | |
| **Colonna** | **Tipo di dato** | **Attributi[[2]](#footnote-2)** |
| **Username** | VARCHAR(45) | PK, NN |
| **Password** | VARCHAR(45) | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella categoria** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| **nome** | VARCHAR(45) | PK, NN |
| **macrocategoria** | VARCHAR(45) |  |

|  |  |  |
| --- | --- | --- |
| **Tabella offerta** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| **utente** | VARCHAR(16) | PK, NN |
| **oggetto\_in\_asta** | VARCHAR(10) | PK, NN |
| **importo** | FLOAT | PK, NN |
| **data\_inserimento** | DATE | NN |
| **ora\_inserimento** | TIME | NN |
| **tipo** | ENUM(‘offerta’,’controfferta automatica’) | NN |
| **importo\_massimo** | FLOAT |  |

|  |  |  |
| --- | --- | --- |
| **Tabella oggetto\_in\_asta** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| **codice** | VARCHAR(10) | PK, NN |
| **descrizione** | VARCHAR(45) | NN |
| **stato** | VARCHAR(45) | NN |
| **prezzo\_di\_base** | FLOAT | NN |
| **descrizione\_dimensioni** | VARCHAR(45) | NN |
| **tipo** | ENUM(‘oggetto in asta’, ’oggetto non venduto’, ‘oggetto venduto’) | NN |
| **numero\_offerte** | INT |  |
| **valore\_massima\_offerta** | FLOAT |  |
| **data\_inizio\_asta** | DATE | NN |
| **data\_fine\_asta** | DATE | NN |
| **orario\_fine\_asta** | TIME | NN |
| **durata\_asta** | INT | NN |
| **categoria** | VARCHAR(45) | NN |
| **prezzo\_di\_vendita** | FLOAT |  |
| **utente** | VARCHAR(16) |  |

|  |  |  |
| --- | --- | --- |
| **Tabella utente** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| **CF** | VARCHAR(16) | PK, NN |
| **nome** | VARCHAR(45) | NN |
| **cognome** | VARCHAR(45) | NN |
| **data\_di\_nascita** | DATE | NN |
| **città\_di\_nascita** | VARCHAR(45) | NN |
| **numero\_carta\_di\_credito** | VARCHAR(16) | NN, UQ |
| **data\_di\_scadenza\_carta\_di\_credito** | DATE | NN |
| **cvv** | VARCHAR(3) | NN, UQ |
| **via\_di\_consegna** | VARCHAR(45) | NN |
| **comune\_di\_consegna** | VARCHAR(45) | NN |
| **cap\_di\_consegna** | VARCHAR(45) | NN |
| **numero\_civico\_di\_consegna** | VARCHAR(45) | NN |
| **username** | VARCHAR(45) | NN, UQ |
| **password** | 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

|  |  |
| --- | --- |
| **Tabella oggetto\_in\_asta** | |
| **Indice data\_fine\_asta** | **Tipo[[3]](#footnote-3):** |
| data\_fine\_asta | IDX |

## L’indice viene inserito in modo tale da rendere più efficiente l’esecuzione delle operazioni U3 e U5, cioè le operazioni che consentono ad un utente di visualizzare le aste aperte e le aste aperte in cui hanno fatto almeno un’offerta.

## Trigger

## Controllo del livello gerarchico del titolario gerarchico delle categorie, tale controllo viene effettuato prima dell'inserimento di una nuova categoria che presenta una macrocategoria.

CREATE DEFINER=`root`@`localhost`

TRIGGER `controllo\_livello\_gerarchico\_categoria\_inserimento`

BEFORE INSERT ON `categoria` FOR EACH ROW

BEGIN

declare livello int;

declare nome\_macrocategoria varchar(45);

set livello = 0;

set nome\_macrocategoria = new.macrocategoria;

while nome\_macrocategoria is not null do

set livello = livello + 1;

select macrocategoria into nome\_macrocategoria

from categoria

where nome = nome\_macrocategoria;

if livello > 2 then

signal sqlstate '45010' set message\_text = "La categoria non può essere inserita, il livello gerarchico è già 3.";

end if;

end while;

END

## Controllo del livello gerarchico delle categorie prima di un aggiornamento di una categoria nella tabella.

CREATE DEFINER=`root`@`localhost`

TRIGGER `controllo\_livello\_gerarchico\_categoria\_aggiornamento`

BEFORE UPDATE ON `categoria` FOR EACH ROW

BEGIN

declare livello int;

declare nome\_macrocategoria varchar(45);

set livello = 0;

set nome\_macrocategoria = new.macrocategoria;

while nome\_macrocategoria is not null do

set livello = livello + 1;

select macrocategoria into nome\_macrocategoria

from categoria

where nome = nome\_macrocategoria;

if livello > 2 then

signal sqlstate '45010' set message\_text = "La categoria non può essere inserita, il livello gerarchico è già 3.";

end if;

end while;

END

* Controllo che la durata dell’asta per un oggetto sia compresa tra 1 e 7 giorni, tale trigger implementa la regola aziendale definita in fase di progettazione concettuale.

CREATE

DEFINER=`root`@`localhost`

TRIGGER `aste\_onlineDB`.`controllo\_durata\_asta`

BEFORE INSERT ON `aste\_onlineDB`.`oggetto\_in\_asta`

FOR EACH ROW

BEGIN

if new.durata\_asta < 1 or new.durata\_asta > 7 then

signal sqlstate '45002' set message\_text = "La durata dell'asta non è compresa tra 1 e 7 giorni";

end if;

END

## Eventi

## Viene aggiunto l’evento relativo alla cancellazione dei dati più vecchi di 5 anni, che risultano trascurabili ai fini dell’applicazione, vengono anche cancellati gli utenti che non hanno fatto alcuna offerta da almeno un anno all’interno del sistema. In questo modo si evita che il volume dei dati mantenuti all’interno del sistema crescano in eccesso mantenendo dati irrilevanti ai fini del funzionamento del sistema. L’evento verrà eseguito per la prima volta dopo 5 anni e verrà ripetuto ogni anno, in modo da tenere il volume dei dati della base di dati a regina. Considerando i vincoli di foreign keys in fase di configurazione del sistema, una volta eliminato un utente verranno eliminati tutti gli oggetti e le offerte che l’utente ha fatto all’interno del sistema in modo tale da non introdurre ulteriori statement di DELETE.

CREATE EVENT IF NOT exists `aste\_onlineDB`.`elimazione\_dati\_vecchi`

on schedule every 1 year starts current\_time + interval 5 year on completion preserve

comment "Rimozione dati più vecchi di 5 anni"

do

begin

delete from utente

where cf in (

select cf from utente

where cf not in (

select cf

from utente as u join offerta as o on o.utente = u.cf

group by data\_inserimento, u.cf

having max(data\_inserimento) >= date\_sub(now(), interval 1 year)

)

);

delete from oggetto\_in\_asta

where codice in (

select codice

from oggetto\_in\_asta

where data\_fine\_asta >= date\_sub(now, interval 5 year)

);

end

## Viene aggiunto l’evento per cambiare lo stato di un’asta per un preciso oggetto in asta, questo avviene cambiando l’attributo tipo all’interno della tabella oggetto\_in\_asta. Questo evento viene eseguito una volta al minuto, questo perché non possiamo prevedere quando un’asta possa terminare.

CREATE EVENT IF NOT exists `aste\_onlineDB`.`modifica\_stato\_oggetto\_in\_asta`

on schedule every 1 minute starts current\_time on completion preserve

comment "implementazione cambio stato di un oggetto in asta al termine dell'asta"

do

begin

declare var\_cf\_utente varchar(16);

declare var\_codice\_oggetto varchar(10);

declare var\_prezzo\_di\_vendita float;

declare var\_numero\_offerte int;

declare var\_ora time(1);

-- cerco i dati relativi agli oggetti in asta per cui l'asta riferita ad essi è terminata

set var\_ora = current\_time();

select codice, numero\_offerte, valore\_massima\_offerta

from oggetto\_in\_asta

where current\_date() > data\_fine\_asta

and tipo = 'oggetto in asta'

union

select codice, numero\_offerte, valore\_massima\_offerta

from oggetto\_in\_asta

where current\_date() = data\_fine\_asta and current\_time() > orario\_inizio\_asta

and tipo = 'oggetto in asta'

limit 1

into var\_codice\_oggetto, var\_numero\_offerte, var\_prezzo\_di\_vendita;

-- se il numero di offerte è 0 allora l'oggetto non è stato venduto

if(var\_numero\_offerte = 0) then

update oggetto\_in\_asta

set tipo = 'oggetto non venduto'

where codice = var\_codice\_oggetto;

elseif(var\_numero\_offerte > 0) then

-- trovo l'utente che ha fatto l'offerta massima

select offerta.utente

from oggetto\_in\_asta join offerta on offerta.oggetto\_in\_asta = oggetto\_in\_asta.codice

where oggetto\_in\_asta.codice = var\_codice\_oggetto and offerta.importo = var\_prezzo\_di\_vendita

into var\_cf\_utente;

-- aggiorno lo stato dell'oggetto

update oggetto\_in\_asta

set tipo = 'oggetto venduto', prezzo\_di\_vendita = var\_prezzo\_di\_vendita,

utente = var\_cf\_utente

where codice = var\_codice\_oggetto;

end if;

end

## Viste

## Non sono state utilizzate viste.

**Funzioni**

* Viene utilizzata la seguente funzione per controllare che il cap dell’indirizzo di consegna venga inserito nel modo corretto in fase di registrazione degli utenti.

CREATE DEFINER=`root`@`localhost` FUNCTION `controllo\_cap`(cap varchar(5)) RETURNS tinyint(1)

deterministic

BEGIN

if cap regexp '[0-9]{5}$' then

return true;

else

return false;

end if;

END

* Viene utilizzata la seguente funzione per controllare che il numero della carta di credito di un utente venga inserito in modo corretto.

CREATE

DEFINER=`root`@`localhost`

FUNCTION `controllo\_carta\_di\_credito`(carta\_di\_credito varchar(16)) RETURNS tinyint(1)

deterministic

BEGIN

if carta\_di\_credito regexp '[0-9]{16}$' then

return true;

else

return false;

end if;

END

* Viene utilizzata la seguente funzione per controllare che il codice fiscale inserito da un utente in fase di registrazione venga inserito in modo corretto.

CREATE DEFINER=`root`@`localhost` FUNCTION `controllo\_cf`(codice\_fiscale varchar(16)) RETURNS tinyint(1)

deterministic

BEGIN

if codice\_fiscale regexp '[A-Z]{6}[0-9]{2}[A-Z][0-9]{2}[A-Z][0-9]{3}[A-Z]$' then

return true;

else

return false;

end if;

END

* Viene utilizzata la seguente funzione per controllare che il cvv della carta di credito venga inserito in modo corretto nella fase di registrazione degli utenti.

CREATE DEFINER=`root`@`localhost` FUNCTION `controllo\_cvv`(cvv varchar(3)) RETURNS tinyint(1)

deterministic

BEGIN

if cvv regexp '[0-9]{3}$' then

return true;

else

return false;

end if;

END

* Viene utilizzata la seguente funzione per controllare che la carta di credito di un utente non sia scaduta, tale controllo viene effettuato in fase di registrazione di un utente. In questo modo viene implementata una prima parte della regola aziendale dove un utente con la carta di credito scaduta NON DEVE poter acquistare oggetti in asta. In questa funzione viene implementata solo una parte di tale regola aziendale, perché in fase di registrazione non possiamo sapere quando un utente effettuerà delle offerte nel sistema.

CREATE DEFINER=`root`@`localhost` FUNCTION `controllo\_scadenza\_carta\_di\_credito`(data\_scadenza date) RETURNS tinyint(1)

deterministic

BEGIN

if(data\_scadenza < current\_date()) then

return false;

else

return true;

end if;

END

## Stored Procedures e transazioni

* Procedura introdotta per eliminare una categoria dal sistema. Viene utilizzato come livello di isolamento Repeatable Read in modo tale da bloccare i lock in lettura e in scrittura per quella specifica tupla all’interno della tabella in modo tale che l’operazione venga eseguita in modo corretto anche nel caso di concorrenza in cui più amministratori siano interessati ad accedere a quella tupla con altre operazioni relative alle categorie.

CREATE DEFINER=`root`@`localhost` PROCEDURE `elimina\_categoria`(

in var\_nome\_categoria varchar(45)

)

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level repeatable read;

start transaction;

delete from categoria

where nome = var\_nome\_categoria;

commit;

END

* Viene introdotta la seguente procedura per l’eliminazione di una macrocategoria da una categoria presente nel sistema. Viene utilizzato come livello di isolamento per la transazione il livello Repeatable Read in modo tale da bloccare i lock in lettura e in scrittura per quella specifica tupla in modo tale da garantire il corretto funzionamento anche in caso di concorrenza dove più amministratori volessero accedere a quella tupla all’interno della tabella.

CREATE DEFINER=`root`@`localhost` PROCEDURE `elimina\_macrocategoria`(

in var\_nome\_categoria varchar(45)

)

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level repeatable read;

start transaction;

update categoria

set macrocategoria = NULL

where nome = var\_nome\_categoria;

commit;

END

* Procedura per far effettuare le offerte agli utenti per gli oggetti in asta. Vengono prima effettuati dei controlli per verificare che il prezzo, nel caso in cui l’offerta che l’utente sta facendo fosse la prima, sia almeno pari al prezzo di base d’asta per quell’oggetto. Viene controllato che l’importo sia multiplo di 0,50 €. Viene controllato che l’importo dell’offerta sia maggiore di tutte le offerte fatte per quello specifico oggetto. In questo modo vengono implementate le regole aziendali relative al prezzo delle offerte individuate in progettazione concettuale. Viene controllato che l’oggetto su cui viene fatta l’offerta sia effettivamente in asta e quindi che un utente possa fare offerte solo su oggetti in asta e non su altri tipi di oggetti. In questo modo viene implementata la regola aziendale. Viene controllato che la carta di credito dell’utente non scada prima della fine dell’asta per quell’oggetto su cui sta facendo un’offerta, in questo modo viene implementata la seconda parte della regola aziendale per cui un utente con la carta di credito scaduta non deve poter acquistare degli oggetti. Una volta fatti tutti i controlli, viene inserita nella tabella offerta l’effettiva offerta fatta dall’utente e viene aggiornato il numero di offerte per quell’oggetto e il valore dell’offerta massima valida per lo stesso oggetto. Dopo l’inserimento si verifica se sia attiva la funzionalità di controfferta automatica per conto di un utente; nel caso in cui questa sia attiva, vengono recuperati i dati relativi all’utente che ha attivato tale funzionalità, si controlla che l’importo impostato dall’utente per la controfferta automatica sia almeno pari al valore dell’importo appena inserito per quell’oggetto più 0,50 € e poi si genera una nuova offerta per conto di quell’utente, che ha attivato la funzionalità di controfferta automatica, aggiungendo all’importo massimo temporaneo valido un valore di 0,50 €, come da specifica; una volta generata la nuova offerta, viene aggiornato il numero di offerte e il valore della massima offerta valida per quell’oggetto. L’utente che viene ricercato, se esiste, è quello che avrà impostato il valore massimo raggiungibile per quell’oggetto in asta massimo, quando genererà la prima offerta valida per quell’oggetto attivando la funzionalità di controfferta automatica imposterà il valore massimo dell’importo che è intenzionato a spendere per quell’oggetto ed lui che sarà potrà vincere l’asta e quindi generare l’offerta con il valore dell’importo massimo nel caso in cui non vengano generate offerte con il valore dell’importo maggiore del valore massimo impostato. Dato che tale operazione impone l’esecuzione di un’interrogazione coinvolgente più righe della tabella, utilizziamo come livello di isolamento il livello Serializable in modo da mantenere i range lock per le righe della tabella.

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

in var\_cf\_utente varchar(16),

in var\_codice\_oggetto varchar(10),

in var\_importo float

)

BEGIN

-- valori per l'offerta da inserire

-- e per i controlli

declare var\_data\_scadenza\_carta date;

declare var\_data\_fine\_asta date;

declare var\_ora\_fine\_asta time(1);

declare tipo\_oggetto enum('oggetto in asta', 'oggetto non venduto', 'oggetto venduto');

declare var\_offerta\_massima float;

declare var\_prezzo\_base\_oggetto float;

declare var\_data\_inserimento\_offerta date;

declare var\_orario\_inserimento\_offerta time(1);

declare var\_numero\_offerte int;

declare var\_nuovo\_numero\_offerte int;

declare var\_tipo\_offerta enum('offerta', 'controfferta automatica');

-- valori della nuova offerta generata da

-- controfferta automatica

declare var\_utente\_controfferta varchar(16);

declare var\_oggetto\_in\_asta\_controfferta varchar(10);

declare var\_importo\_massimo\_controfferta float;

declare var\_ora\_inserimento\_controfferta time(1);

declare var\_data\_inserimento\_controfferta date;

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level serializable;

start transaction;

-- controllo i valori di inserimento

select prezzo\_di\_base, numero\_offerte

into var\_prezzo\_base\_oggetto, var\_numero\_offerte

from oggetto\_in\_asta

where codice = var\_codice\_oggetto;

if(var\_numero\_offerte = 0) then

if(var\_importo < var\_prezzo\_base\_oggetto) then

signal sqlstate '45003' set message\_text = "L'offerta non è valida, è la prima offerta e deve essre almeno pari al prezzo iniziale di base d'asta";

end if;

end if;

if(var\_importo % 0.50 <> 0) then

SIGNAL SQLSTATE '45004' SET MESSAGE\_TEXT = 'Il valore deve essere un multiplo di 0.50.';

end if;

select valore\_massima\_offerta into var\_offerta\_massima

from oggetto\_in\_asta

where codice = var\_codice\_oggetto;

if(var\_importo <= var\_offerta\_massima) then

SIGNAL SQLSTATE '45005' SET MESSAGE\_TEXT = "Il valore deve essere maggiore di tutte le offerte per l'oggetto specificato.";

end if;

select data\_fine\_asta, orario\_inizio\_asta, tipo

from oggetto\_in\_asta

where codice = var\_codice\_oggetto

into var\_data\_fine\_asta, var\_ora\_fine\_asta, tipo\_oggetto;

if(tipo\_oggetto <> 'oggetto in asta') then

signal sqlstate '45059' set message\_text = "Non puoi fare offerte su questo oggetto.";

end if;

if(var\_data\_fine\_asta < current\_date()) then

signal sqlstate '45059' set message\_text = "Non puoi fare offerte su questo oggetto.";

end if;

if(var\_data\_fine\_asta = current\_date() and var\_ora\_fine\_asta < current\_time()) then

signal sqlstate '45059' set message\_text = "Non puoi fare offerte su questo oggetto.";

end if;

select data\_di\_scadenza\_carta\_di\_credito

from utente

where cf = var\_cf\_utente

into var\_data\_scadenza\_carta;

if(var\_data\_scadenza\_carta < var\_data\_fine\_asta) then

signal sqlstate '45089' set message\_text = "Non puoi fare un'offerta su questo oggetto, la tua carta di credito scade prima della fine dell'asta.";

end if;

-- inzializzo i dati relativi all'offerta appena inserita

set var\_data\_inserimento\_offerta = current\_date();

set var\_orario\_inserimento\_offerta = current\_time();

set var\_tipo\_offerta = 'offerta';

select numero\_offerte into var\_numero\_offerte

from oggetto\_in\_asta

where codice = var\_codice\_oggetto;

set var\_nuovo\_numero\_offerte = var\_numero\_offerte + 1;

-- inserisco la nuova offerta

insert into offerta

(utente, oggetto\_in\_asta, importo, data\_inserimento,

ora\_inserimento, tipo)

values (var\_cf\_utente, var\_codice\_oggetto, var\_importo,

var\_data\_inserimento\_offerta, var\_orario\_inserimento\_offerta,

var\_tipo\_offerta);

-- aggiorno il numero delle offerte per l'oggetto

update oggetto\_in\_asta

set numero\_offerte = var\_nuovo\_numero\_offerte,

valore\_massima\_offerta = var\_importo

where codice = var\_codice\_oggetto;

-- trovo l'utente e il valore dell'importo dell'offerta automatica

set var\_oggetto\_in\_asta\_controfferta = var\_codice\_oggetto;

select utente, importo\_massimo

from offerta

where oggetto\_in\_asta = var\_oggetto\_in\_asta\_controfferta

and tipo = 'controfferta automatica'

and utente <> var\_cf\_utente

group by utente, importo\_massimo

having max(importo\_massimo)

order by importo\_massimo desc

limit 1

into var\_utente\_controfferta, var\_importo\_massimo\_controfferta;

if(var\_utente\_controfferta is not null and

(var\_importo + 0.50) <= var\_importo\_massimo\_controfferta) then

-- inizializzo i parametri per generare la nuova offerta

set var\_ora\_inserimento\_controfferta = var\_orario\_inserimento\_offerta;

set var\_data\_inserimento\_controfferta = current\_date();

-- genero la nuova offerta

insert into offerta

(utente, oggetto\_in\_asta, importo, data\_inserimento,

ora\_inserimento, tipo, importo\_massimo)

values

(var\_utente\_controfferta, var\_oggetto\_in\_asta\_controfferta,

var\_importo + 0.50, var\_data\_inserimento\_controfferta,

var\_ora\_inserimento\_controfferta, 'controfferta automatica',

var\_importo\_massimo\_controfferta);

-- aggiorno i valori nella tabella oggetto\_in\_asta

update oggetto\_in\_asta

set numero\_offerte = numero\_offerte + 1,

valore\_massima\_offerta = var\_importo + 0.50

where codice = var\_oggetto\_in\_asta\_controfferta;

end if;

commit;

END

* Procedura per attivare la funzionalità di controfferta automatica per un utente. Tale procedura inizialmente fa gli stessi controlli relativi ai valori passati dall’utente in input; quali il valore dell’offerta che sia multiplo di 0,50 €, che sia maggiore del valore del prezzo di base d’asta nel caso in cui sia la prima offerta per quell’oggetto e che sia maggiore del valore dell’importo massimo dell’offerta maggiore per quell’oggetto. Vengono fatti i controlli per verificare se l’utente possa fare delle offerte su quell’oggetto; quindi, che l’oggetto non sia stato venduto o non venduto e viene controllato se la carta di credito dell’utente non scada prima della fine dell’asta. Dopo aver fatto tutti i controlli, viene inserita la nuova offerta con il valore dell’importo massimo che utente vuole spendere su quell’oggetto, si aggiornano i dati relativi al numero di offerte e al valore dell’importo del valore della massima offerta attuale per quell’oggetto in asta. Una volta inserita l’offerta, viene controllato se esiste un utente che ha già impostato la funzionalità di controfferta automatica, nel caso in cui tale valore sia maggiore del valore dell’importo massimo appena impostato, si controlla che tale importo sia almeno pari al valore dell’importo dell’offerta più 0,50 € e viene generata la nuova offerta per poi aggiornare il numero dell’offerta e il valore dell’importo della massima offerta su quell’oggetto in asta. Dato che nella ricerca dell’utente che ha attivato la funzionalità di controfferta automatica che abbia il valore più grande relativo all’importo massimo che ha impostato attivando la funzionalità di controfferta si effettua un’interrogazione su un insieme di tuple, si utilizza come livello di isolamento Serializable.

CREATE DEFINER=`root`@`localhost` PROCEDURE `fai\_prima\_offerta\_controfferta\_automatica`(

in var\_cf\_utente varchar(16),

in var\_codice\_oggetto varchar(10),

in var\_importo\_offerta\_iniziale float,

in valore\_importo\_massimo float

)

BEGIN

declare var\_data\_scadenza\_carta date;

declare var\_data\_fine\_asta date;

declare var\_ora\_fine\_asta time(1);

declare tipo\_oggetto enum('oggetto in asta', 'oggetto non venduto', 'oggetto venduto');

declare var\_offerta\_massima float;

declare var\_prezzo\_base\_oggetto float;

declare var\_tipo\_offerta enum('offerta', 'controfferta automatica');

declare var\_data\_inserimento date;

declare var\_orario\_inserimento time(1);

declare var\_numero\_offerte int;

declare var\_nuovo\_numero\_offerte int;

-- valori della nuova offerta generata da

-- controfferta automatica

declare var\_utente\_controfferta varchar(16);

declare var\_oggetto\_in\_asta\_controfferta varchar(10);

declare var\_importo\_massimo\_controfferta float;

declare var\_ora\_inserimento\_controfferta time(1);

declare var\_data\_inserimento\_controfferta date;

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level serializable;

start transaction;

select prezzo\_di\_base, numero\_offerte

into var\_prezzo\_base\_oggetto, var\_numero\_offerte

from oggetto\_in\_asta

where codice = var\_codice\_oggetto;

if(var\_numero\_offerte = 0) then

if(var\_importo\_offerta\_iniziale < var\_prezzo\_base\_oggetto) then

signal sqlstate '45003' set message\_text = "L'offerta non è valida, è la prima offerta e deve essre almeno pari al prezzo iniziale di base d'asta";

end if;

end if;

if(var\_importo\_offerta\_iniziale % 0.50 <> 0) then

SIGNAL SQLSTATE '45004' SET MESSAGE\_TEXT = 'Il valore deve essere un multiplo di 0.50.';

end if;

select valore\_massima\_offerta into var\_offerta\_massima

from oggetto\_in\_asta

where codice = var\_codice\_oggetto;

if(var\_importo\_offerta\_iniziale <= var\_offerta\_massima) then

SIGNAL SQLSTATE '45005' SET MESSAGE\_TEXT = "Il valore deve essere maggiore di tutte le offerte per l'oggetto specificato.";

end if;

select data\_fine\_asta, orario\_inizio\_asta, tipo

from oggetto\_in\_asta

where codice = var\_codice\_oggetto

into var\_data\_fine\_asta, var\_ora\_fine\_asta, tipo\_oggetto;

if(tipo\_oggetto <> 'oggetto in asta') then

signal sqlstate '45059' set message\_text = "Non puoi fare offerte su questo oggetto.";

end if;

if(var\_data\_fine\_asta < current\_date()) then

signal sqlstate '45059' set message\_text = "Non puoi fare offerte su questo oggetto.";

end if;

if(var\_data\_fine\_asta = current\_date() and var\_ora\_fine\_asta < current\_time()) then

signal sqlstate '45059' set message\_text = "Non puoi fare offerte su questo oggetto.";

end if;

select data\_di\_scadenza\_carta\_di\_credito

from utente

where cf = var\_cf\_utente

into var\_data\_scadenza\_carta;

if(var\_data\_scadenza\_carta < var\_data\_fine\_asta) then

signal sqlstate '45089' set message\_text = "Non puoi fare un'offerta su questo oggetto, la tua carta di credito scade prima della fine dell'asta.";

end if;

set var\_tipo\_offerta = 'controfferta automatica';

set var\_data\_inserimento = current\_date();

set var\_orario\_inserimento = current\_time();

select numero\_offerte into var\_numero\_offerte

from oggetto\_in\_asta

where codice = var\_codice\_oggetto;

set var\_nuovo\_numero\_offerte = var\_numero\_offerte + 1;

insert into offerta

(utente, oggetto\_in\_asta, importo, data\_inserimento,

ora\_inserimento, tipo, importo\_massimo)

values (var\_cf\_utente, var\_codice\_oggetto, var\_importo\_offerta\_iniziale,

var\_data\_inserimento, var\_orario\_inserimento, var\_tipo\_offerta,

valore\_importo\_massimo);

update oggetto\_in\_asta

set numero\_offerte = var\_nuovo\_numero\_offerte,

valore\_massima\_offerta = var\_importo\_offerta\_iniziale

where codice = var\_codice\_oggetto;

-- trovo l'utente e il valore dell'importo dell'offerta automatica

set var\_oggetto\_in\_asta\_controfferta = var\_codice\_oggetto;

select utente, importo\_massimo

from offerta

where oggetto\_in\_asta = var\_oggetto\_in\_asta\_controfferta

and tipo = 'controfferta automatica'

and utente <> var\_cf\_utente

group by utente, importo\_massimo

having max(importo\_massimo) > valore\_importo\_massimo

order by importo\_massimo desc

limit 1

into var\_utente\_controfferta, var\_importo\_massimo\_controfferta;

if(var\_utente\_controfferta is not null and

(var\_importo\_offerta\_iniziale + 0.50) <= var\_importo\_massimo\_controfferta) then

-- inizializzo i parametri per generare la nuova offerta

set var\_ora\_inserimento\_controfferta = var\_orario\_inserimento;

set var\_data\_inserimento\_controfferta = current\_date();

-- genero la nuova offerta

insert into offerta

(utente, oggetto\_in\_asta, importo, data\_inserimento,

ora\_inserimento, tipo, importo\_massimo)

values

(var\_utente\_controfferta, var\_oggetto\_in\_asta\_controfferta,

var\_importo\_offerta\_iniziale + 0.50, var\_data\_inserimento\_controfferta,

var\_ora\_inserimento\_controfferta, 'controfferta automatica',

var\_importo\_massimo\_controfferta);

-- aggiorno i valori nella tabella oggetto\_in\_asta

update oggetto\_in\_asta

set numero\_offerte = numero\_offerte + 1,

valore\_massima\_offerta = var\_importo\_offerta\_iniziale + 0.50

where codice = var\_oggetto\_in\_asta\_controfferta;

end if;

commit;

END

* Procedura utilizzata per l’inserimento di una categoria che ha una macrocategoria.

CREATE DEFINER=`root`@`localhost` PROCEDURE `inserisci\_categoria\_con\_macrocategoria`(

in var\_nome\_categoria varchar(45),

in var\_nome\_macrocategoria varchar(45)

)

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level repeatable read;

start transaction;

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

commit;

END

* Procedura utilizzata per l’inserimento di una categoria che non ha una macrocategoria.

CREATE DEFINER=`root`@`localhost` PROCEDURE `inserisci\_categoria\_senza\_macrocategoria`(

in var\_nome\_categoria varchar(45)

)

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read uncommitted;

start transaction;

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

commit;

END

* Procedura utilizzata per inserire un oggetto in asta.

CREATE DEFINER=`root`@`localhost` PROCEDURE `inserisci\_oggetto\_in\_asta`(

in var\_descrizione varchar(45),

in var\_stato varchar(45),

in var\_prezzo\_di\_base float,

in var\_descrizione\_dimensioni varchar(45),

in var\_durata\_asta int,

in var\_categoria varchar(45)

)

BEGIN

declare var\_codice\_oggetto varchar(10);

declare var\_tipo enum ('oggetto in asta', 'oggetto non venduto', 'oggetto venduto');

declare var\_numero\_offerte int;

declare var\_data\_inizio\_asta date;

declare var\_data\_fine\_asta date;

declare var\_orario\_inizio\_asta time(1);

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read uncommitted;

start transaction;

set var\_codice\_oggetto = substring(replace(uuid(), '-', ''), 1, 10);

set var\_tipo = 'oggetto in asta';

set var\_numero\_offerte = 0;

set var\_data\_inizio\_asta = current\_date();

set var\_data\_fine\_asta = date\_add(var\_data\_inizio\_asta, interval var\_durata\_asta day);

set var\_orario\_inizio\_asta = current\_time();

insert into oggetto\_in\_asta

(codice, descrizione, stato, prezzo\_di\_base, descrizione\_dimensioni,

tipo, numero\_offerte, data\_inizio\_asta, durata\_asta, data\_fine\_asta,

orario\_inizio\_asta, categoria)

values

(var\_codice\_oggetto, var\_descrizione, var\_stato, var\_prezzo\_di\_base,

var\_descrizione\_dimensioni, var\_tipo, var\_numero\_offerte, var\_data\_inizio\_asta,

var\_durata\_asta, var\_data\_fine\_asta, var\_orario\_inizio\_asta, var\_categoria);

commit;

END

* Procedura che per mette il login di un utente o di un amministratore.

CREATE DEFINER=`root`@`localhost` PROCEDURE `login`(in var\_username

VARCHAR(45), in var\_password VARCHAR(45), out var\_role INT)

BEGIN

-- var\_role = 3 --> login

-- var\_role = 1 --> utente

-- var\_role = 2 --> amministratore

set var\_role = 3;

if exists(

select \* from utente

WHERE username = var\_username AND

password = md5(var\_password))

then set var\_role = 1;

end if;

if exists(

select \* from amministratori

WHERE username = var\_username AND

password = md5(var\_password))

then set var\_role = 2;

end if;

END

* Procedura utilizzata per la modifica di una macrocategoria di una categoria presente nel sistema. Viene utilizzato il livello di isolamento Repeatable Read per prendere i lock in lettura e in scrittura affinché l’esecuzione dell’operazione avvenga correttamente, questo perché più amministratori possono poter accedere alle stesse righe della tabella.

CREATE DEFINER=`root`@`localhost` PROCEDURE `modifica\_macrocategoria`(

in var\_nome\_categoria varchar(45),

in var\_nome\_macrocategoria varchar(45)

)

BEGIN

declare exit handler for sqlexception

* begin

rollback;

resignal;

end;

set transaction isolation level repeatable read;

start transaction;

update categoria

set macrocategoria = var\_nome\_macrocategoria

where nome = var\_nome\_categoria;

commit;

END

* Procedura utilizzata per modificare il nome di una categoria. Viene utilizzato come livello di isolamento il livello Repeatable Read per prendere i lock in lettura e in scrittura affinché l’esecuzione dell’operazione avvenga nel modo corretto, questo perché più amministratori potrebbero voler accedere a quelle righe della tabella.

CREATE DEFINER=`root`@`localhost` PROCEDURE `modifica\_nome\_categoria`(

in var\_vecchio\_nome varchar(45),

in var\_nuovo\_nome varchar(45)

)

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level repeatable read;

start transaction;

update categoria

set nome = var\_nuovo\_nome

where nome = var\_vecchio\_nome;

commit;

END

* Procedura utilizzata per ottenere tutte le categorie, e le relative macrocategorie se presenti, nel sistema. Si utilizza come livello di isolamento il livello di Read Committed affinché un amministratore possa vedere dati aggiornati che non contengano valori incorretti, quindi per evitare le letture sporche.

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

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read committed;

start transaction;

select nome, macrocategoria

from categoria;

commit;

END

* Procedura utilizzata per riprendere i dati dell’utente che sta utilizzando il sistema.

CREATE DEFINER=`root`@`localhost` PROCEDURE `ottieni\_utente\_by\_username`(in var\_username varchar(45))

BEGIN

select cf, nome, cognome, data\_di\_nascita,

città\_di\_nascita, numero\_carta\_di\_credito,

data\_di\_scadenza\_carta\_di\_credito, cvv, via\_di\_consegna,

comune\_di\_consegna, cap\_di\_consegna, username, password,

numero\_civico\_di\_consegna

from utente

where username = var\_username;

END

* Procedura utilizzata per la registrazione degli utenti. L’operazione viene eseguita in transazione per avere la certezza di avere un’esecuzione atomica e indivisibile considerando i controlli da fare per i dati passati in input dall’utente. Dato che non si soffre di nessuna anomalia, si sceglie di utilizzare come livello di isolamento Read Uncommitted.

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

in var\_cf varchar(16),

in var\_nome varchar(45),

in var\_cognome varchar(45),

in var\_data\_di\_nascita date,

in var\_città\_di\_nascita varchar(45),

in var\_numero\_carta\_di\_credito varchar(16),

in var\_data\_di\_scadenza\_carta\_di\_credito date,

in var\_cvv varchar(3),

in var\_via\_di\_consegna varchar(45),

in var\_numero\_civico\_di\_consegna varchar(45),

in var\_comune\_di\_consegna varchar(45),

in var\_cap\_di\_consegna varchar(45),

in var\_username varchar(45),

in var\_password varchar(45)

)

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read uncommitted;

start transaction;

if(`aste\_onlineDB`.controllo\_cf(var\_cf) is false) then

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

end if;

if(`aste\_onlineDB`.controllo\_carta\_di\_credito(var\_numero\_carta\_di\_credito) is false) then

signal sqlstate '45006' set message\_text = "Il numero della carta di credito non è valido.";

end if;

if(`aste\_onlineDB`.controllo\_scadenza\_carta\_di\_credito(var\_data\_di\_scadenza\_carta\_di\_credito) is false) then

signal sqlstate '45031' set message\_text = 'La carta di credito è scaduta.';

end if;

if(`aste\_onlineDB`.controllo\_cvv(var\_cvv) is false) then

signal sqlstate '45007' set message\_text = "Il valore di cvv inserito non è valido.";

end if;

if(`aste\_onlineDB`.controllo\_cap(var\_cap\_di\_consegna) is false) then

signal sqlstate '45030' set message\_text = "Il cap inserito non è valido.";

end if;

insert into utente

(cf, nome, cognome, data\_di\_nascita, città\_di\_nascita, numero\_carta\_di\_credito,

data\_di\_scadenza\_carta\_di\_credito, cvv, via\_di\_consegna, comune\_di\_consegna, cap\_di\_consegna,username, password, numero\_civico\_di\_consegna)

values (var\_cf, var\_nome, var\_cognome, var\_data\_di\_nascita,

var\_città\_di\_nascita, var\_numero\_carta\_di\_credito, var\_data\_di\_scadenza\_carta\_di\_credito,

var\_cvv, var\_via\_di\_consegna, var\_comune\_di\_consegna,

var\_cap\_di\_consegna, var\_username, md5(var\_password), var\_numero\_civico\_di\_consegna

);

commit;

END

* Procedura in grado di far registrare gli amministratori al sistema.

CREATE DEFINER=`root`@`localhost` PROCEDURE `registrazione\_amministratori`(in var\_username varchar(45), in var\_password varchar(45))

BEGIN

insert into amministratori (username, password) values (var\_username, md5(var\_password));

END

* Procedura utilizzata per vedere le aste aperte e gli oggetti a cui sono riferite.

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

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read uncommitted;

start transaction;

select codice, descrizione, stato, descrizione\_dimensioni,

prezzo\_di\_base, numero\_offerte, data\_fine\_asta,

orario\_inizio\_asta, valore\_massima\_offerta, categoria

from oggetto\_in\_asta

where data\_fine\_asta > current\_date()

union

select codice, descrizione, stato, descrizione\_dimensioni,

prezzo\_di\_base, numero\_offerte, data\_fine\_asta,

orario\_inizio\_asta, valore\_massima\_offerta, categoria

from oggetto\_in\_asta

where data\_fine\_asta = current\_date() and

orario\_inizio\_asta - current\_time() > 0;

commit;

END

* Procedura utilizzata dagli utenti per vedere gli oggetti che hanno acquistato.

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

in var\_cf\_utente varchar(16)

)

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read uncommitted;

start transaction;

select descrizione, stato,

descrizione\_dimensioni as 'descrizione dimensioni', categoria,

prezzo\_di\_vendita as 'prezzo di vendita'

from oggetto\_in\_asta

where utente = var\_cf\_utente;

commit;

END

* Procedura utilizzata dagli utenti per vedere lo stato delle aste in cui hanno fatto almeno un’offerta.

CREATE DEFINER=`root`@`localhost` PROCEDURE `visualizza\_stato\_aste\_con\_offerte`(

in var\_cf\_utente varchar(45)

)

BEGIN

declare exit handler for sqlexception

begin

rollback;

resignal;

end;

set transaction isolation level read uncommitted;

start transaction;

select codice, descrizione, stato,

descrizione\_dimensioni as 'descrizione dimensioni',

prezzo\_di\_base as 'prezzo di base',

data\_fine\_asta as 'data fine asta',

orario\_inizio\_asta as 'orario inizio asta',

valore\_massima\_offerta as 'valore massima offerta'

from oggetto\_in\_asta join offerta o on

oggetto\_in\_asta.codice = o.oggetto\_in\_asta

where data\_fine\_asta > current\_date() and

oggetto\_in\_asta.tipo = 'oggetto in asta' and

var\_cf\_utente in (select utente from offerta

where offerta.utente = o.utente)

union

select codice, descrizione, stato,

descrizione\_dimensioni as 'descrizione dimensioni',

prezzo\_di\_base as 'prezzo di base',

data\_fine\_asta as 'data fine asta',

orario\_inizio\_asta as 'orario inizio asta',

valore\_massima\_offerta as 'valore massima offerta'

from oggetto\_in\_asta join offerta o

on oggetto\_in\_asta.codice = o.oggetto\_in\_asta

where data\_fine\_asta = current\_date() and

oggetto\_in\_asta.tipo = 'oggetto in asta' and

orario\_inizio\_asta - current\_time() > 0

and var\_cf\_utente in (select utente from offerta

where offerta.utente = o.utente);

commit;

END

1. Indicare con E le entità, con R le relazioni [↑](#footnote-ref-1)
2. 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-2)
3. IDX = index, UQ = unique, FT = full text, PR = primary. [↑](#footnote-ref-3)