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

Basi di Dati

Progetto A.A. 2022/2023

Openbay

0270289

Giuliano Vallone

**Indice**

[1. Descrizione del Minimondo 3](#_Toc606296459)

[2. Analisi dei Requisiti 4](#_Toc1289394997)

[3. Progettazione concettuale 5](#_Toc2081466291)

[4. Progettazione logica 6](#_Toc2147004904)

[5. Progettazione fisica 8](#_Toc518560220)

[Appendice: Implementazione 9](#_Toc403811585)

# 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 | 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 ciascun oggetto 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** |
| 11-12 | Da una descrizione delle dimensioni | Dalle dimensioni | È già presente il termine descrizione (linea. 10) che indica un altro concetto. |
| 14 | Asta | Oggetto | Il concetto di categoria viene associato all’asta a riga 14, mentre ad Oggetto nella frase successiva. Ragionevolmente, dato che ogni asta è legata ad un singolo oggetto, concettualmente è l’Oggetto ad appartenere ad una categoria. |
| 15 | Titolario gerarchico | Gerarchia a livelli |  |

### 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. 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 dalle 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 associata una categoria. Le categorie appartengono ad una gerarchia a livelli, organizzata 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. |

## Glossario dei Termini

|  |  |  |  |
| --- | --- | --- | --- |
| **Termine** | **Descrizione** | **Sinonimi** | **Collegamenti** |
| Utenti | Coloro che utilizzano il sistema per partecipare alle aste |  | Asta, Carta di credito |
| Carta di credito | Carta utilizzata per effettuare i pagamenti delle aste vinte |  | Utenti |
| Oggetto | Oggetto messo all’asta |  | Utenti, Asta |
| Asta | Asta di un oggetto in uno specifico intervallo di tempo |  | Oggetto, Utenti |
| Offerta | Offerta fatta da un utente per un oggetto specifico |  | Oggetto, Controfferta, Utenti |
| Controfferta | Offerta effettuata automaticamente dal sistema per un utente, nel momento in cui un utente fa un’offerta più alta |  | Utenti, Offerta |
| Amministratori | Coloro che gestiscono il sistema. |  | Asta, Oggetto |

## Raggruppamento dei requisiti in insiemi omogenei

|  |
| --- |
| **Frasi relative a Utenti** |
| Gli utenti del sistema, previa registrazione, hanno la possibilità di fare offerte su un qualsiasi oggetto.  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 ... .  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.  Inoltre, un utente che ha attualmente piazzato l’offerta massima, può sfruttare la funzionalità di “controfferta automatica”.  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 Carta di credito** |
| ... carta di credito (intestatario, numero, data di scadenza, codice CVV). |

|  |
| --- |
| **Frasi relative a Oggetto** |
| 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 dalle dimensioni.  Inoltre, a ciascun oggetto viene associata una categoria. Le categorie appartengono ad una gerarchia a livelli, organizzata su un massimo di tre livelli. |

|  |
| --- |
| **Frasi relative a Asta** |
| Al termine dell’asta, l’offerta maggiore sarà quella che avrà vinto l’asta.  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. |

|  |
| --- |
| **Frasi relative a Offerta** |
| La granularità di incremento delle offerte è di multipli di 50 centesimi di euro.  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. |

|  |
| --- |
| **Frasi relative a Controfferta** |
| Inoltre, un utente che ha attualmente piazzato l’offerta massima, può sfruttare la funzionalità di “controfferta automatica”.  Tale funzionalità permette all’utente di indicare un importo massimo con cui si intende rilanciare l’offerta, qualora un altro utente faccia un’offerta maggiore.  La gestione delle offerte pertanto funziona nel modo seguente. L’utente A indica un importo I con cui vuole rilanciare l’offerta nei confronti dell’utente B che è attualmente il migliore offerente. L’utente B ha anche indicato un importo di controfferta C. Se C > I, il sistema indicherà come miglior offerente l’utente A, con importo temporaneo I, ma immediatamente dopo indicherà nuovamente l’utente B come migliore offerente, con un importo di I + 0,50€.  Ciò significa che tutte le transazioni automatiche generate dal sistema di controfferta automatica devono essere registrate nel sistema. |

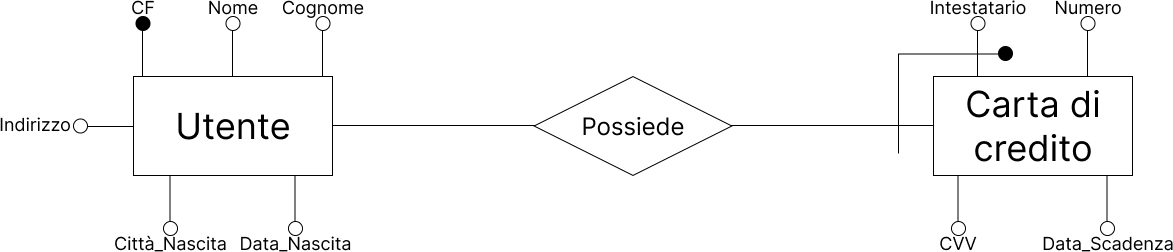
|  |
| --- |
| **Frasi relative a Amministratori** |
| Il sistema deve consentire agli amministratori la gestione degli oggetti che si vogliono pubblicare e tutto il ciclo di vita delle aste.  Gli amministratori gestiscono l’inserimento degli oggetti.  Quando viene inserito un nuovo oggetto nel sistema, gli amministratori possono decidere la durata dell’asta ... .  La gestione delle categorie degli oggetti afferisce sempre agli amministratori del sistema. |

# Progettazione concettuale

## Costruzione dello schema E-R

Per la costruzione dello schema E-R ho adottato una progettazione mista tra una strategia a macchia d’olio e una bottom-up.

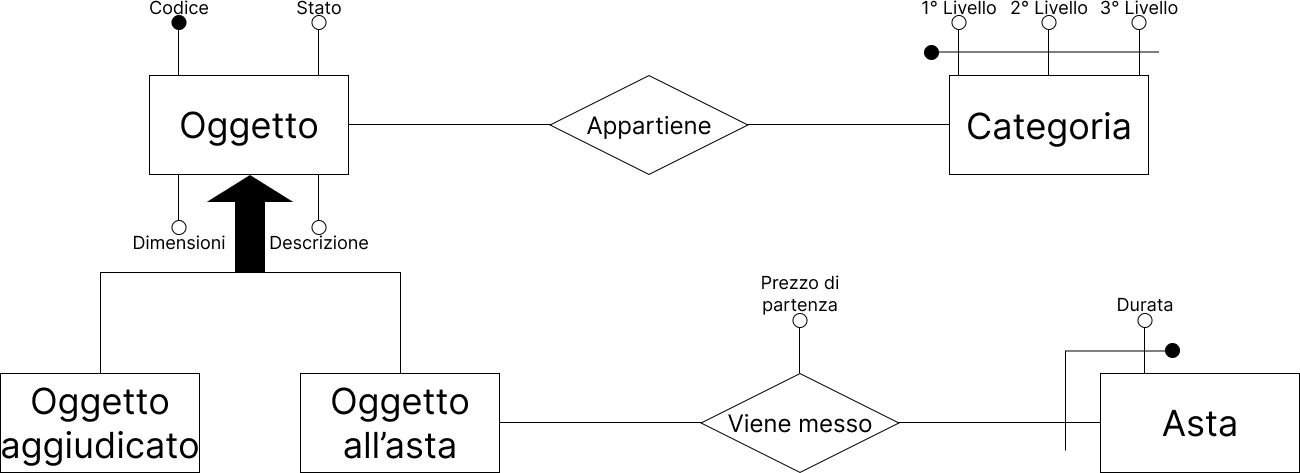
Come primo passo ho deciso di fare sviluppare i dati sugli utenti e le carte di credito, assumendo che ogni utente inserisca una sola carta di credito.



Schema 3.1

Come identificatori ho scelto per l’entità Utente il Codice Fiscale (rappresentato nello schema 3.1 come CF), mentre per l’entità Carta di Credito ho scelto di utilizzare come chiave esterna l’Intestatario.

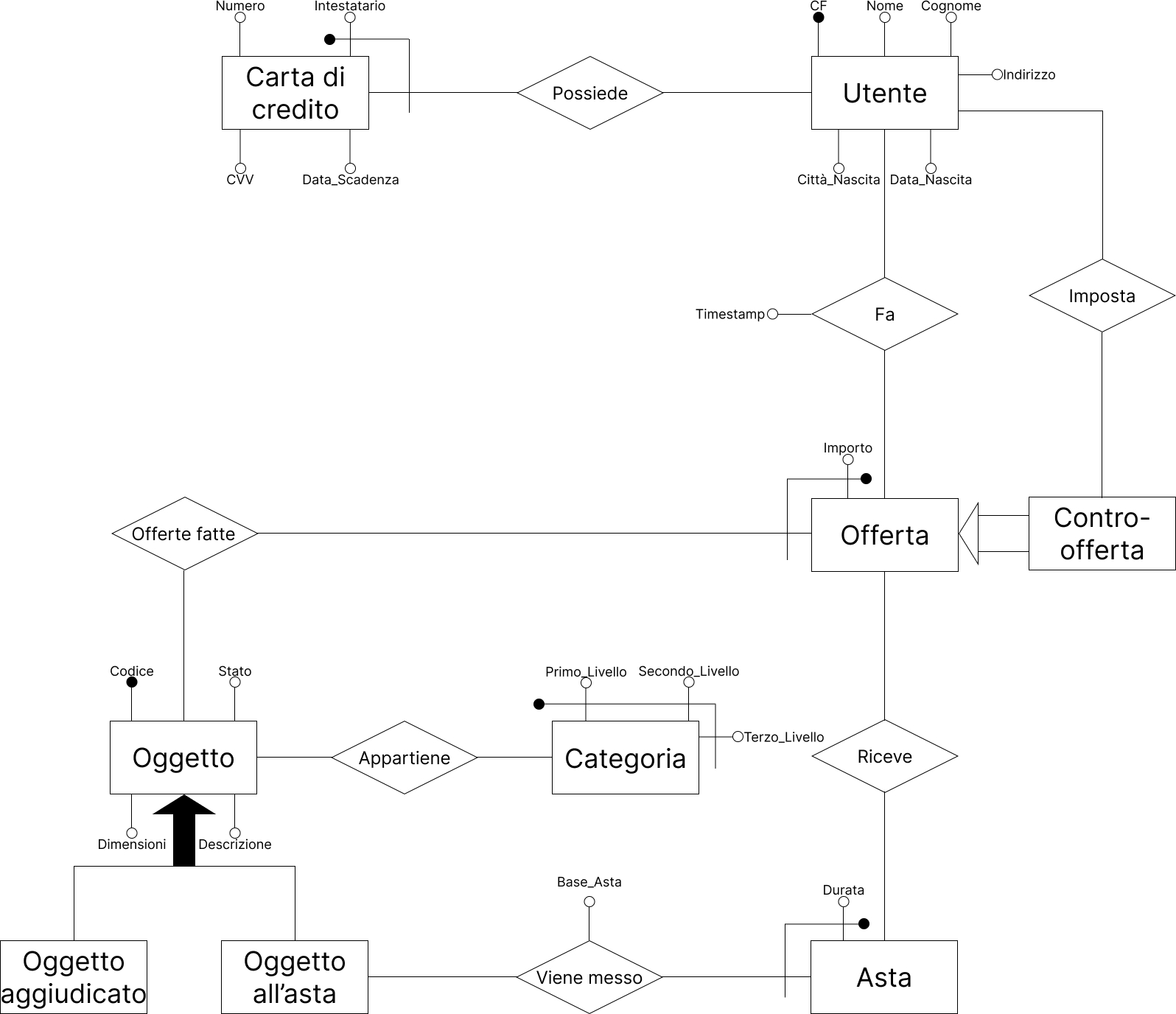
La seconda parte che ho sviluppato è stata relativa alle Aste e agli Oggetti, separandoli in Oggetto venduto e Oggetto all’asta.



Schema 3.2

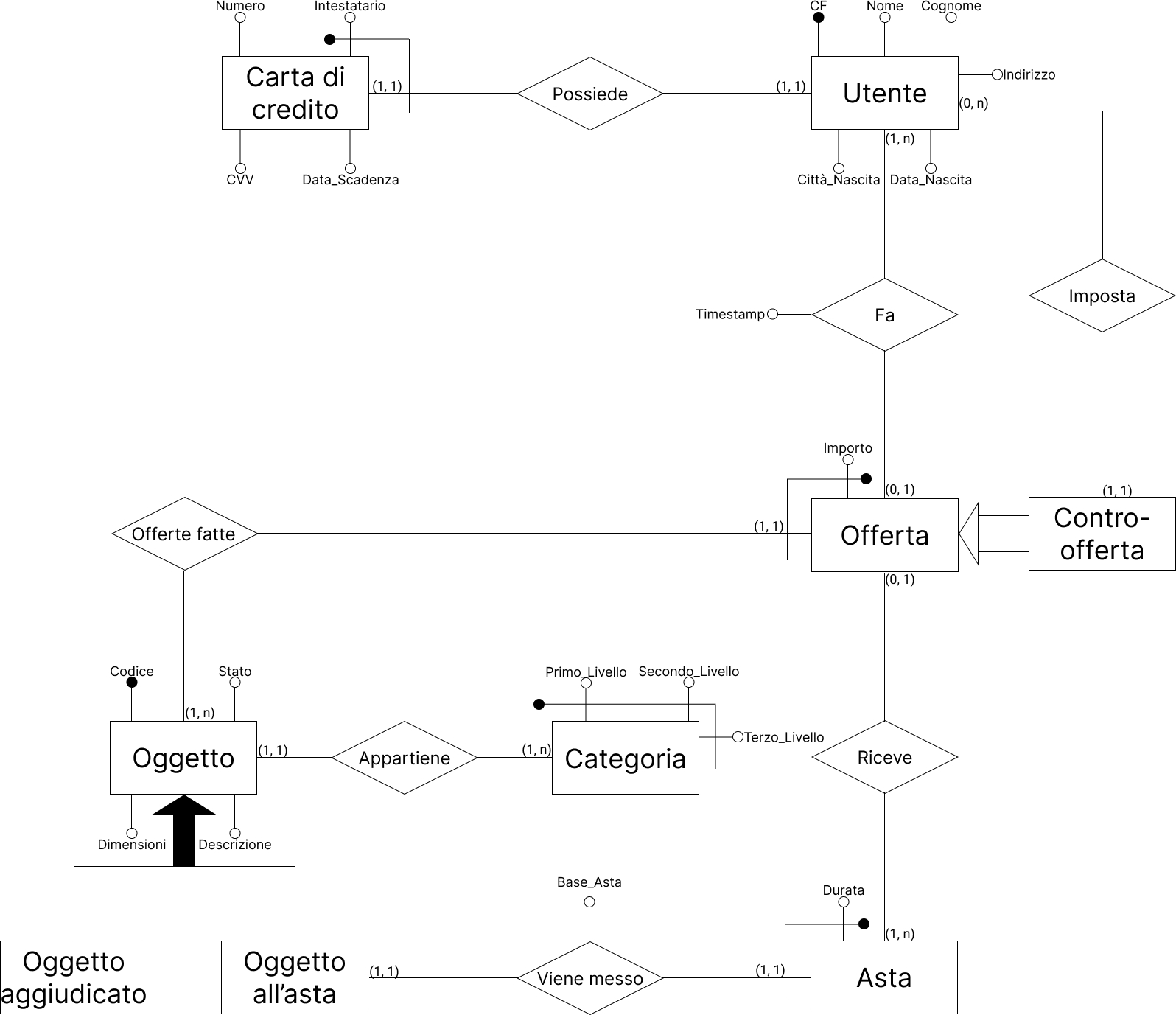
Ho deciso di rappresentare le categorie come entità piuttosto che come attributo composto in quanto nelle specifiche è richiesto che gli amministratori del sistema le gestiscano esplicitamente. Come identificatore di Oggetto ho scelto l’attributo Codice, in quanto nelle specifiche è esplicitato che sia univoco. Per l’entità Asta invece ho scelto di inserire una chiave esterna in quanto l’indicitura dell’asta, da specifica, è dipendente dall’inserimento di un nuovo oggetto. Per quanto riguarda l’entità Categoria ho scelto di utilizzare una chiave composta in quanto due categorie di primo livello diverse fra loro potrebbero contenere due categorie di secondo livello con lo stesso nome (questo discorso è applicabile anche tra categorie di secondo e terzo livello).

Infine ho effettuato l’integrazione tra gli schemi 3.1 e 3.2 tramite le Offerte e la loro storicizzazione.



Schema 3.3

Infine ho inserito nello schema 3.4 le cardinalità delle relazioni, ottenendo infine lo schema finale



Schema 3.4

### Integrazione finale

Nell’integrazione finale non ho trovato conflitti strutturali o sui nomi, quindi l’integrazione finale è quella riportata nello schema 3.4 .

## Regole aziendali

* La durata di un’asta deve essere da 1 a 7 giorni;
* L’incremento di un offerta deve essere un multiplo di €0,50;
* Primo, secondo e terzo livello di una categoria devono avere ognuna un nome diverso;
* Il codice degli oggetti è composta da 3 lettere e 3 numeri separati da un trattino (es. AAA-111) che verrà generato casualmente dal sistema.
* Il sistema tiene salvata solo la controfferta dell’ultimo “controfferente”,

## Dizionario dei dati

|  |  |  |  |
| --- | --- | --- | --- |
| **Entità** | **Descrizione** | **Attributi** | **Identificatori** |
| Utente | Persona che fa le offerte agli oggetti all’asta. | CF, Nome, Cognome, Città\_Nascita, Data\_Nascita, Indirizzo | CF |
| Carta di credito | Carta con cui gli utenti effettueranno il pagamento alla vincita di un’asta. | Intestatario. Numero, CVV, Data\_Scadenza | Intestatario, CF |
| Offerte | Insieme di offerte effettuate da un utente nel per una specifica asta. | Timestamp, Euro | Timestamp, CF |
| Controfferta | Offerta automatica effettuata dal maggior offerente in risposta ad un’offerta fatta da un altro utente | Timestamp, Euro | Timestamp, CF |
| Asta | Asta di un oggetto | Durata | Durata, Codice |
| Oggetto | Insieme degli oggetti che sono/sono stati messi all’asta | Codice, Stato, Dimensioni, Descrizione, Categoria | Codice |
| Oggetto aggiudicato | Oggetto aggiudicato | Codice, Stato, Dimensioni, Descrizione, Categoria | Codice |
| Oggetto all’asta | Oggetto attualmente all’asta non ancora aggiudicato | Codice, Stato, Dimensioni, Descrizione, Categoria | Codice |
| Categoria | Categoria a cui appartiene un oggetto definita secondo una gerarchia a tre livelli | Primo\_Livello, Secondo\_Livello, Terzo\_Livello | Primo\_Livello, Secondo\_Livello, Terzo\_Livello |

# Progettazione logica

## Volume dei dati

|  |  |  |
| --- | --- | --- |
| **Concetto nello schema** | **Tipo[[1]](#footnote-2)** | **Volume atteso** |
| Utente | E | 10000 |
| Carta di credito | E | 10000 |
| Offerta | E | 1000000 |
| Controfferta | E | 10000 |
| Asta | E | 100 |
| Oggetto all’asta | E | 100 |
| Oggetto | E | 10000 |
| Oggetto aggiudicato | E | 9900 |
| Categoria | E | 100 |
| Possiede | R | 10000 |
| Fa | R | 1000000 |
| Riceve | R | 10000 |
| Viene messo | R | 100 |
| Offerte fatte | R | 1000000 |
| Appartiene | R | 10000 |
| Imposta | R | 10000 |

## Tavola delle operazioni

|  |  |  |
| --- | --- | --- |
| **Cod.** | **Descrizione** | **Frequenza attesa** |
| U01 | Fai un’offerta su un oggetto | 10000/settimana |
| U02 | Imposta Controfferta | 1000/settimana |
| U03 | Visualizza lo stato delle aste in corso a cui un utente partecipa | 10000/settimana |
| U04 | Visualizza elenco degli oggetti all’asta | 1000/settimana |
| U05 | Visualizza elenco degli oggetti aggiudicati | 1000/settimana |
| A01 | Inserisci oggetto e indici l’asta | 200/settimana |
| A02 | Inserisci categoria | 10/anno |
| L01 | Login | 4000/settimana |
| L02 | Registrazione | 10/mese |

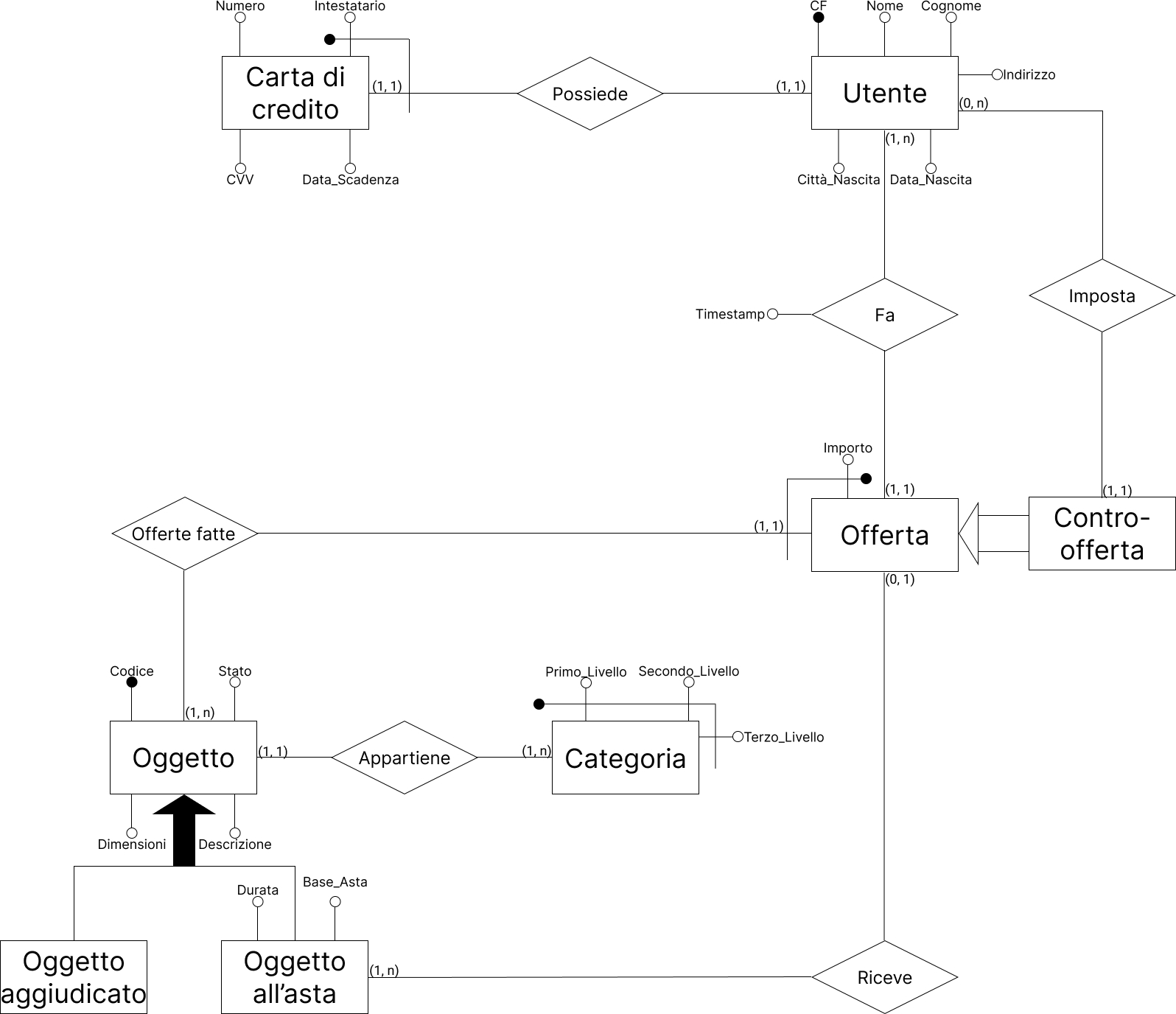
## Costo delle operazioni

* Operazione U01:
  + 1 accesso in scrittura sulla relazione Fa;
  + 1 accesso in scrittura su Offerte;
  + 1 accesso in lettura su Controfferta;
  + 1 accesso in scrittura su Riceve;
  + Costo = [2\*(1+1+1)+1]\*10000 = 70000 accessi/settimana.
* Operazione U02:
  + 1 accesso in scrittura su Imposta
  + 1 accesso in scrittura su Controfferta
  + Costo = 2\*1000 = 2000
* Operazione U03:
  + 1 accesso in lettura su Fa;
  + 1 accesso in lettura su Offerta;
  + 50 accessi in lettura su Riceve (per calcolare il numero di offerte);
  + 50 accessi in lettura su Riceve (per calcolare l’offerta massima);
  + 1 accesso in lettura su Asta;
  + 1 accesso in lettura su Viene Messo;
  + Costo = (1+1+1+1+50+50)\*10000 = 104\*100000 = 1040000 accessi/settimana.
* Operazione U04:
  + 100 accessi in lettura su Asta;
  + 10000 accessi in lettura su Riceve (per calcolare il numero di offerte)
  + 10000 accessi in lettura su Riceve (per calcolare l’offerta massima)
  + 100 accessi in lettura su Viene Messo;
  + 100 accessi in lettura su Oggetto all’asta;
  + 100 accessi in lettura su Oggetto;
  + 100 accessi in lettura su Appartiene;
  + Costo = (100\*5+10000\*2)\*4000 = 82000000
* Operazione U05:
  + 990000 accessi in lettura su Offerte Fatte; (per calcolare il numero di offerte)
  + 990000 accessi in lettura su Offerte Fatte; (per calcolare l’offerta massima)
  + 9900 accessi in lettura su Oggetti;
  + 9900 accessi in lettura su Appartiene;
  + Costo = (9900\*2+990000\*2)\* = 199980000 accessi/settimana
* Operazione A01:
  + 1 accesso in scrittura su Oggetto;
  + 1 accesso in scrittura su Appartiene;
  + 1 accesso in scrittura su Oggetto all’asta;
  + 1 accesso in scrittura su Viene Messo;
  + 1 accesso in scrittura su Asta;
  + Costo = (1+1+1+1+1)\*2\*200/settimana = 2000 accessi/settimana
* Operazione A02:
  + 1 accesso in scrittura su Categoria:
  + Costo = 1\*2\*10/anno = 20 accessi/anno
* Operazione L02:
  + 1 accesso in scrittura su Utente;
  + 1 accesso in scrittura su Carta di Credito;
  + 1 accesso in scrittura su Possiede;
  + Costo = (1+1+1)\*2\*10/mese = 60 accessi/mese.

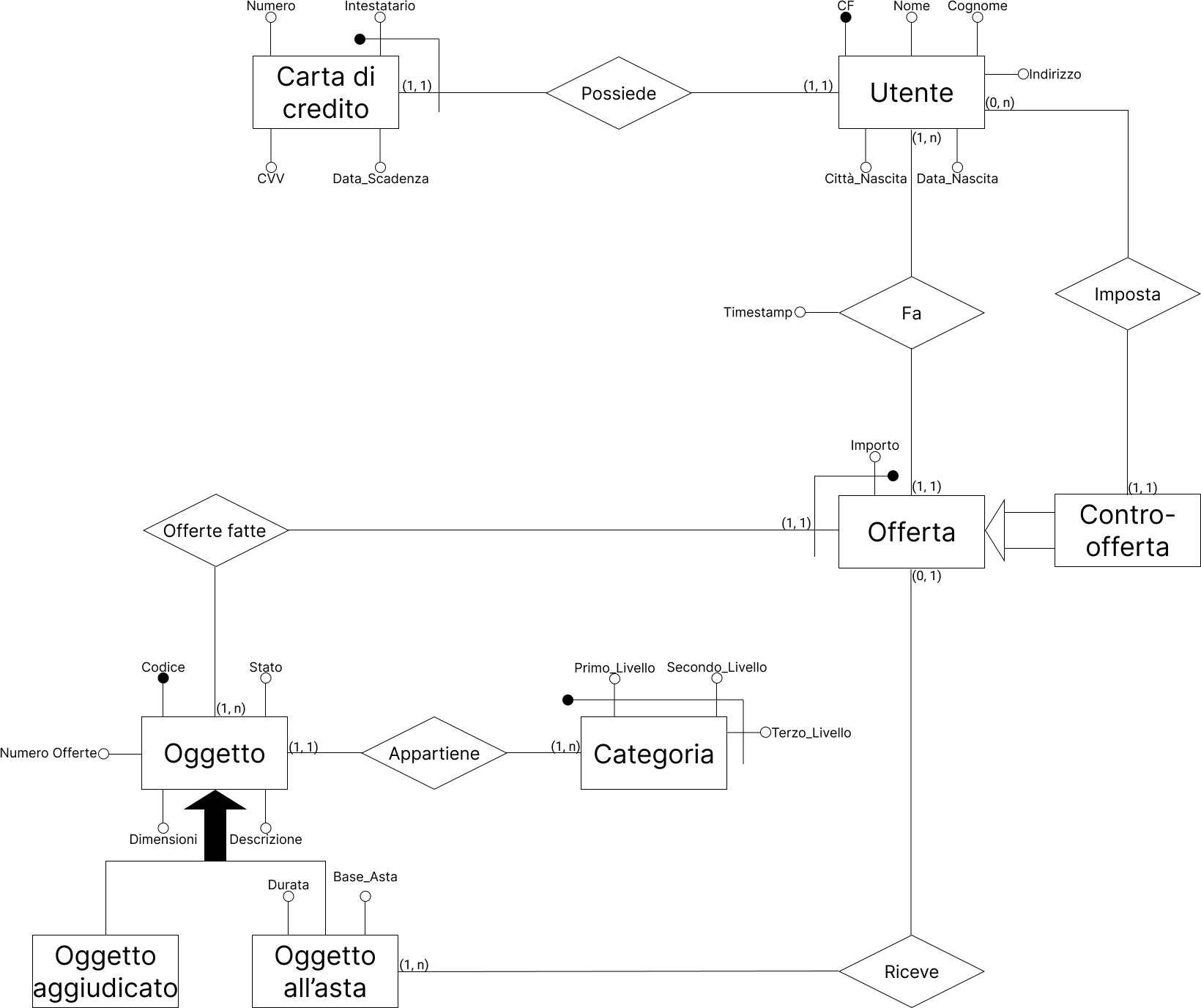
## Ristrutturazione dello schema E-R

Per ristrutturare lo schema E-R ho fatto riferimento principalmente alle offerte U03, U04 e U05, in quanto sono le più costose a livello prestazionale. Ho ristrutturato lo schema E-R facendo questi 3 passi:

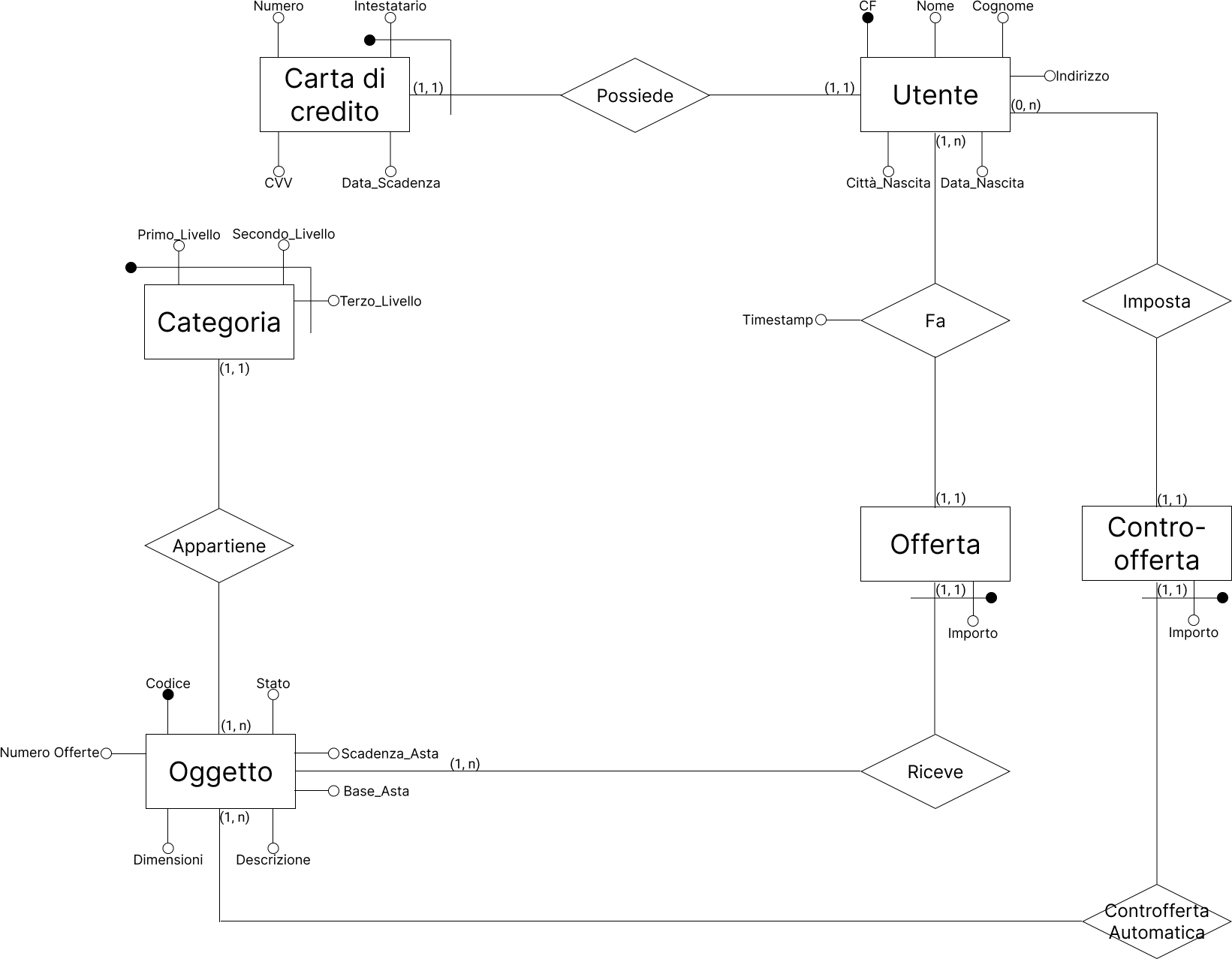
1. Accorpare l’ entità Asta negli Oggetto all’asta;



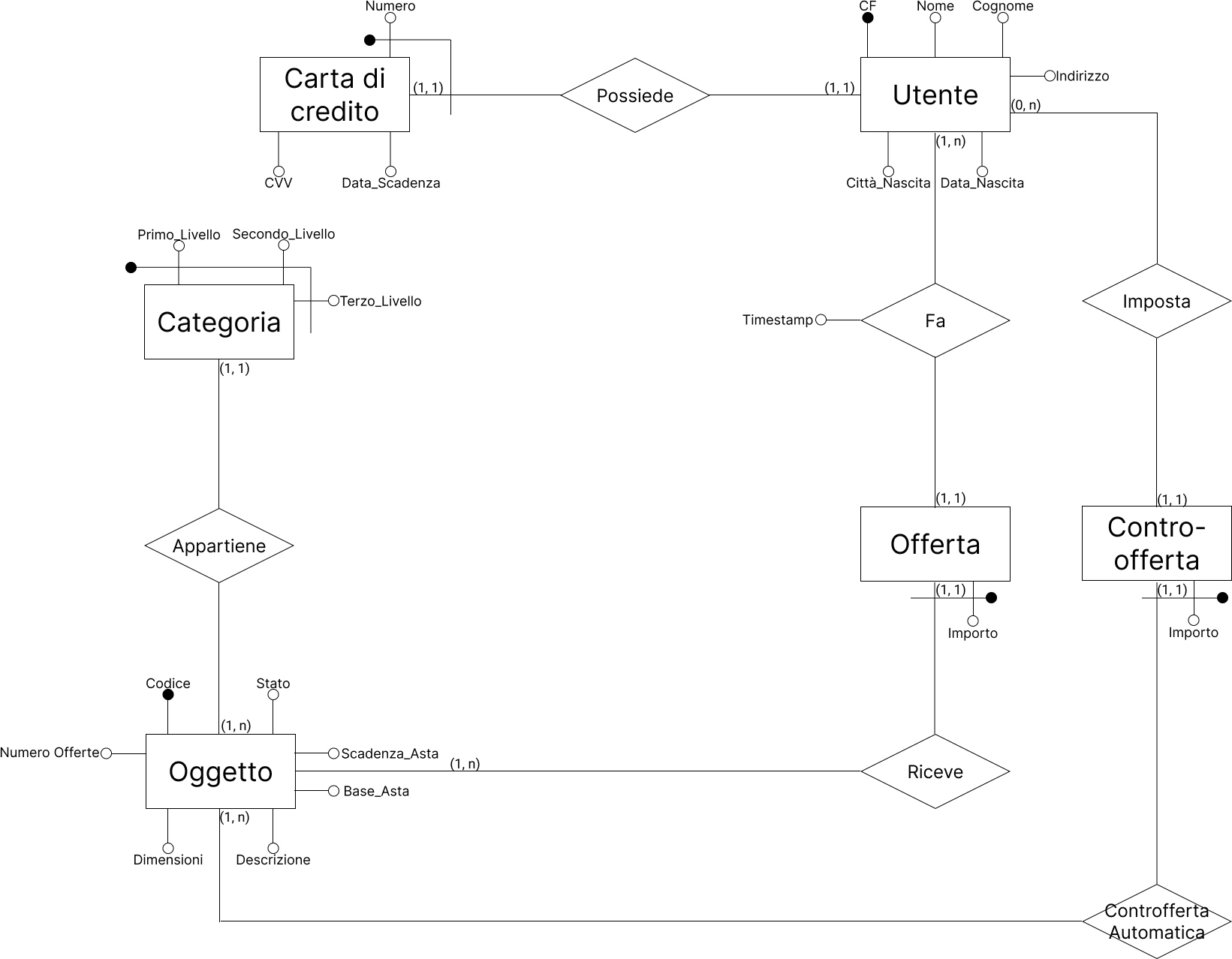
1. Ho aggiunto l’attributo Numero di Offerte all’entità Oggetto all’asta;



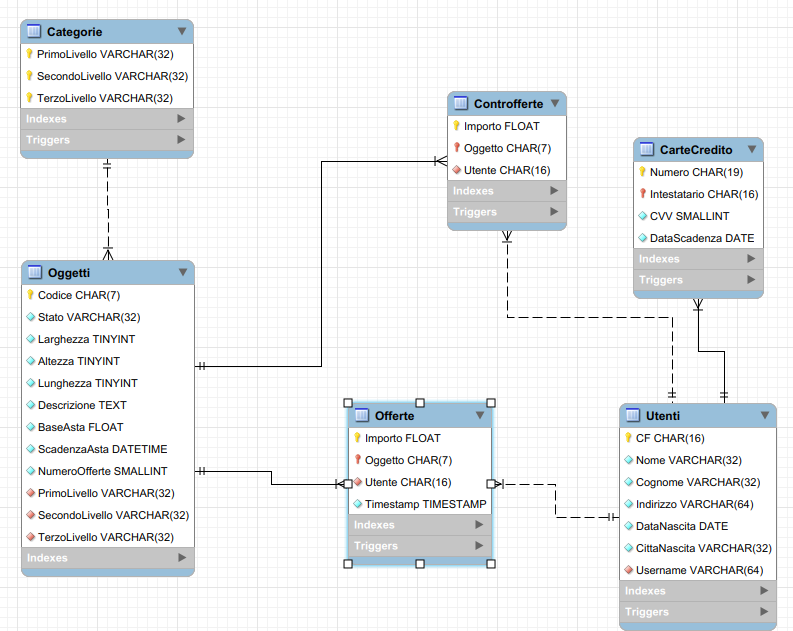
1. Unito le entità Oggetto all’asta e Oggetto aggiudicato nell’entità padre Oggetto, aggiungendo l’attributo Scadenza per differenziarle, eliminando l’attributo Durata.



Infine ho eliminato l’attributo Intestatario (in quanto ridondante) e eliminato la generalizzazione tra l’entità Offerta e l’entità figlia Controfferta.



## Traduzione di entità e associazioni



## Normalizzazione del modello relazionale

Il modello NON è in forma normale poiché l’attributo `NumeroOfferte` è derivabile dalla cardinalità della tabella Offerte, inoltre l’aggiunta di quest’attributo provoca anomalie di inserimento, aggiornamento ed eliminazione. Tuttavia comporta un risparmio prestazionale elevato, di conseguenza ho deciso di lasciare il modello relazionale in questa forma.

# Progettazione fisica

## Utenti e privilegi

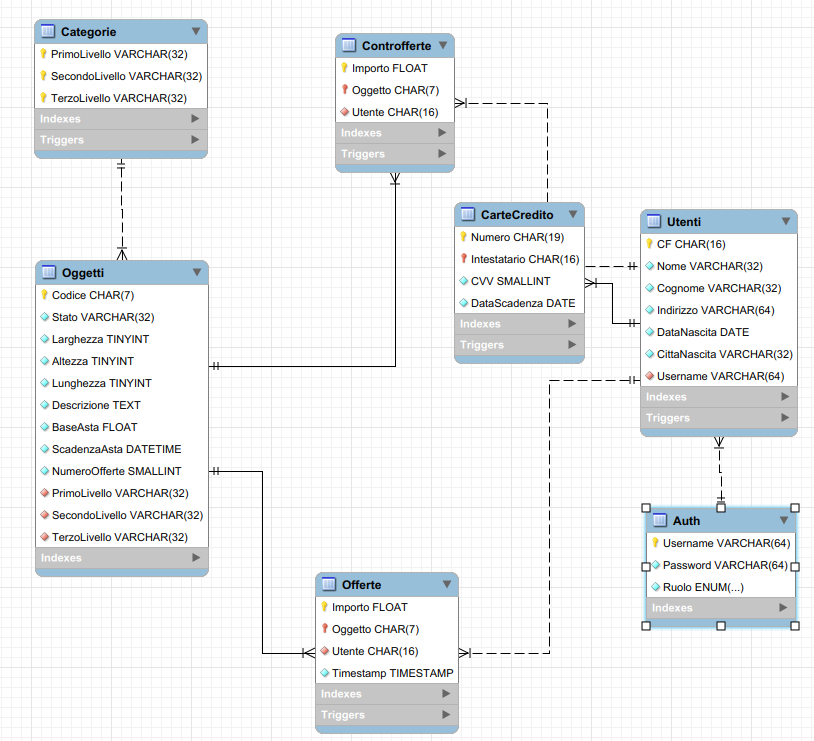
All’interno della base di dati sono stati previsti tre ruoli:

* Login (definito come role\_login);
* Utente (definito come role\_user);
* Amministratore (definito come role\_admin).

I privilegi degli utenti sono stati dati sull’esecuzione stored procedures nel modo seguente:

* Login: L01, L02
* Utente: U01, U02, U03, U04, U05
* Amministratore: A01, A02

Per realizzare la funzionalità di login e la registrazione è stata create la tabella Auth e inserito un riferimento all’attributo ‘Username’ all’interno della tabella ‘Utente’.



## Strutture di memorizzazione

|  |  |  |
| --- | --- | --- |
| **Tabella Auth** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| Username | VARCHAR(32) | PK |
| Password | VARCHAR(32) | NN |
| Role | Enum('amministratore','utente') | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella Carte di credito** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| Numero | CHAR(19) | PK |
| Intestatario | CHAR(16) | PK |
| CVV | SMALLINT | NN, UN |
| Data di scadenza | DATE | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella Categorie** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| 1° Livello | VARCHAR(32) | PK |
| 2° Livello | VARCHAR(32) | PK |
| 3° Livello | VARCHAR(32) | PK |

|  |  |  |
| --- | --- | --- |
| **Tabella Controfferte** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| Importo | FLOAT | PK, UN |
| Utente | CHAR(16) | PK |
| Oggetto | CHAR(7) | PK |

|  |  |  |
| --- | --- | --- |
| **Tabella Offerte** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| Timestamp | TIMESTAMP | PK |
| Utente | CHAR(16) | PK |
| Importo | FLOAT | NN, UN |
| Oggetto | CHAR(7) | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella Oggetti** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| Codice | CHAR(7) | PK |
| Stato | VARCHAR(32) | NN |
| Dimensioni | VARCHAR(32) | NN |
| Descrizione | BLOB | NN |
| Scadenza asta | DATETIME | NN |
| Numero di offerte | SMALLINT | NN, UN |
| Prezzo di partenza | FLOAT | NN, UN |
| 1° Livello | VARCHAR(32) | NN |
| 2° Livello | VARCHAR(32) | NN |
| 3° Livello | VARCHAR(32) | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella Utenti** | | |
| **Colonna** | **Tipo di dato** | **Attributi** |
| CF | CHAR(16) | PK |
| Nome | VARCHAR(32) | NN |
| Cognome | VARCHAR(32) | NN |
| Indirizzo | VARCHAR(64) | NN |
| Data di nascita | DATE | NN |
| Città di nascita | VARCHAR(32) | NN |
| Username | VARCHAR(32) | NN |

## Indici

Dato che vi sono più operazioni che utilizzano l’attributo ‘ScadenzaAsta’ per le ricerche (operazioni U03, U04, U05), ho aggiunto quest’indice per migliorare le prestazioni.

|  |  |
| --- | --- |
| **Tabella Oggetti** | |
| **Indice** ScadenzaAsta | **Tipo:** | |
| ScadenzaAsta | IDX | |

## Trigger

I seguenti trigger servono principalmente per simulare gli assert sui valori degli input.

**Tabella CarteCredito: BEFORE INSERT**

CREATE TRIGGER `assert\_card` BEFORE INSERT ON `CarteCredito`

FOR EACH ROW BEGIN

IF NOT New.Numero REGEXP "^([0-9]{4}-[0-9]{4}-[0-9]{4}-[0-9]{4})" THEN

SIGNAL SQLSTATE "45001" SET MESSAGE\_TEXT = "[InputError] The credit must be kind of XXXX-XXXX-XXXX-XXXX where X is a number!";

END IF;

IF New.CVV > 10000 OR NEW.CVV < 100 THEN

SIGNAL SQLSTATE "45001" SET MESSAGE\_TEXT =

"[InputError] The CVV must a number of digits between 3 and 4!";

END IF;

END

**Tabella Categorie: BEFORE INSERT**

CREATE TRIGGER `assert\_level\_names` BEFORE INSERT ON `Categorie`

FOR EACH ROW BEGIN

IF NEW.`PrimoLivello` = NEW.`SecondoLivello` OR NEW.`PrimoLivello` = NEW.`TerzoLivello` OR NEW.`SecondoLivello` = NEW.`TerzoLivello` THEN

SIGNAL SQLSTATE '45001' SET MESSAGE\_TEXT = "[InputError] The levels's names must be all different!";

END IF;

END

**Tabella Controfferte: BEFORE INSERT**

Il seguente trigger, oltre a verificare che l’input sia corretto, verifica anche che l’utente sia il maggior offerente. Inoltre (dato che il sistema tiene traccia solo dell’ultima controfferta impostata) rimuove un’eventuale controfferta, ormai superata da un’offerta, impostata su quell’oggetto.

CREATE TRIGGER `assert\_counteroffer` BEFORE INSERT ON `Controfferte`

FOR EACH ROW BEGIN

DECLARE increment FLOAT UNSIGNED;

DECLARE importo\_controfferta FLOAT UNSIGNED;

DECLARE max\_offerer CHAR(16);

SET max\_offerer = (SELECT Utente FROM Offerte Where Oggetto = New.Oggetto GROUP BY Oggetto HAVING MAX\_OFFER(Oggetto));

IF NEW.Utente != max\_offerer THEN

SIGNAL SQLSTATE "45002" SET MESSAGE\_TEXT = "[OfferError] You are not the max offerer!";

END IF;

SET importo\_controfferta = (SELECT Importo FROM Controfferte WHERE Oggetto = New.Oggetto);

IF Importo\_controfferta <= MAX\_OFFER(New.Oggetto) THEN

SIGNAL SQLSTATE "45001" SET MESSAGE\_TEXT = "[InputError] The counteroffer must be higher then max offer";

END IF;

SET increment = CAST(((NEW.Importo-offerta\_max)) AS DECIMAL)%0.5;

IF increment != 0.0 THEN

SIGNAL SQLSTATE "45001" SET MESSAGE\_TEXT = "[InputError] The increment must a multiple of 0.50€!";

END IF;

DELETE FROM Controfferte WHERE Oggetto=NEW.Oggetto;

END

**Tabella Offerte: BEFORE INSERT**

Il seguente trigger verifica, oltre alla correttezza dei valori in ingresso, anche che l’utente NON sia il maggior offerente. Inoltre, se vi è una controfferta su quel determinato oggetto, fa il confronto tra l’importo della nuova offerta e della controfferta, facendo l’inserimento secondo le regole scritte nelle specifiche.

CREATE TRIGGER `assert\_offer` BEFORE INSERT ON `Offerte`

FOR EACH ROW BEGIN

DECLARE offerta\_max FLOAT UNSIGNED;

DECLARE increment FLOAT UNSIGNED;

DECLARE max\_offerer CHAR(16);

SET offerta\_max = MAX\_OFFER(New.Oggetto);

SET increment = (CAST(((NEW.Importo-offerta\_max)) AS DECIMAL)%0.5);

IF new.Importo < offerta\_max THEN

SIGNAL SQLSTATE '45002' SET MESSAGE\_TEXT = "Your offer is too low!";

END IF;

IF increment != 0.0 THEN

SIGNAL SQLSTATE '45001' SET MESSAGE\_TEXT = "[InputError] The increment must be a multiple of 0.50€";

END IF;

SET max\_offerer = (SELECT Utente FROM Offerte Where Oggetto = New.Oggetto GROUP BY Oggetto HAVING MAX\_OFFER(Oggetto));

IF NEW.Utente = max\_offerer THEN

SIGNAL SQLSTATE "45002" SET MESSAGE\_TEXT = "[OfferError] You are already the max offerer!";

END IF;

END

**Tabella Utenti: BEFORE INSERT**

CREATE TRIGGER `assert\_cf` BEFORE INSERT ON `Utenti`

FOR EACH ROW BEGIN

IF new.cf REGEXP '^(([A-Z]|[a-z]){6}[0-9]{2}([A-Z]|[a-z])[0-9]{2}([A-Z]|[a-z])[0-9]{3}([A-Z]|[a-z]))' THEN

SIGNAL SQLSTATE "45001" set MESSAGE\_TEXT = "[InputError] The CF inserted is not valid!";

END IF;

END

**Tabella Offerte: AFTER INSERT**

Quest’ultimo trigger implementato sulla tabella ‘Offerte’ è stato inserito per mantenere coerenza tra l’attributo `NumeroOfferte` nella tabella `Oggetti` e il numero di record relative all’oggetto.

CREATE TRIGGER `increment\_offers\_number` AFTER INSERT ON `Offerte`

FOR EACH ROW BEGIN

SET @old\_numero\_offerte := (SELECT NumeroOfferte FROM Oggetti WHERE Codice=New.Oggetto);

UPDATE Oggetti SET NumeroOfferte = @old\_numero\_offerte+1 WHERE Codice=New.Oggetto;

END

## Eventi

Non sono stati implementati eventi.

## Viste

In questa base di dati è stata implementata la seguente vista:

VIEW `Aste` AS

SELECT

`Oggetti`.`Codice` AS `Codice`,

`Oggetti`.`Stato` AS `Stato`,

`Oggetti`.`Lunghezza` AS `Lunghezza`,

`Oggetti`.`Larghezza` AS `Larghezza`,

`Oggetti`.`Altezza` AS `Altezza`,

`Oggetti`.`Descrizione` AS `Descrizione`,

`Oggetti`.`BaseAsta` AS `BaseAsta`,

`Oggetti`.`ScadenzaAsta` AS `ScadenzaAsta`,

`Oggetti`.`NumeroOfferte` AS `NumeroOfferte`,

`Oggetti`.`PrimoLivello` AS `PrimoLivello`,

`Oggetti`.`SecondoLivello` AS `SecondoLivello`,

`Oggetti`.`TerzoLivello` AS `TerzoLivello`,

MAX\_OFFER(`Oggetti`.`Codice`) AS `OffertaMassima`,

`Offerte`.`Utente` AS `Partecipante`

FROM

(`Oggetti`

LEFT JOIN `Offerte` ON ((`Oggetti`.`Codice` = `Offerte`.`Oggetto`)))

## Stored Procedures e transazioni

Mostrare e commentare le stored procedure che sono state realizzate per implementare la logica applicativa delle operazioni sui dati, evidenziando quando (e perché) sono state realizzate operazioni transazionali complesse.

Prima di mostrare le stored procedures riporto il codice SQL della funzione MAX\_OFFER.

**Funzione MAX\_OFFER:**

Ho implementato questa routine perché nel caso non sia stato effettuata alcuna offerta su un determinato oggetto la massima offerta su di esso (non essendoci record) è pari a NULL (provocando problemi con il confronto con questa informazione). Nel momento in cui viene indetta un’asta si potrebbe considerare come massima offerta l’attributo ‘BaseAsta’ della tabella `Oggetti’, questo finché non viene inserita la prima offerta sull’oggetto. Questo inoltre ha semplificato di molto la realizzazione di triggers e stored\_procedures.

CREATE FUNCTION `MAX\_OFFER`(`codice` CHAR(7)) RETURNS float unsigned

DETERMINISTIC

BEGIN

DECLARE numero\_offerte SMALLINT UNSIGNED;

DECLARE base\_asta FLOAT UNSIGNED;

DECLARE offerta\_massima FLOAT UNSIGNED;

SELECT NumeroOfferte, BaseAsta, MAX(Offerte.Importo)

FROM Oggetti LEFT JOIN Offerte ON Oggetti.Codice = Offerte.Oggetto

WHERE Oggetti.Codice = codice INTO numero\_offerte, base\_asta, offerta\_massima;

IF offerta\_massima IS NULL THEN

return base\_asta;

ELSE

return offerta\_massima;

END IF;

END

**Operazione L01:**

CREATE PROCEDURE `login`(IN var\_username varchar(64), IN password varchar(64), OUT var\_role TINYINT(1), OUT var\_cf CHAR(16))

BEGIN

DECLARE user\_role ENUM('amministratore', 'utente');

SELECT `Ruolo`

FROM Auth

WHERE Auth.username = var\_username AND Auth.Password = md5(password) INTO user\_role;

SELECT CF

FROM Utenti

WHERE Username = var\_username INTO var\_cf;

IF user\_role = 'amministratore' THEN

SET var\_role = 1;

ELSEIF user\_role = 'utente' THEN

SET var\_role = 2;

ELSE

SET var\_role = 3;

END if;

END

**Operazione L02:**

Nella seguente operazione ho definito una transazione con livello di isolamento ‘read committed’

per evitare la ‘lettura sporca’, in quanto nel momento in cui un utente A fa la registrazione contemporaneamente all’utente B che ha inserito lo stesso username dell’utente A, la cui registrazione è fallita, il sistema potrebbe leggere lo username dell’utente B.

CREATE PROCEDURE `user\_registration`(IN `username` VARCHAR(32), IN `password`

VARCHAR(32), IN `cf` CHAR(16), IN `nome` VARCHAR(32), IN `cognome` VARCHAR(32), IN `indirizzo` VARCHAR(64), IN `data\_nascita` DATE, IN `citta\_nascita` VARCHAR(32), IN `numero\_carta` CHAR(19), IN `cvv` SMALLINT, IN `data\_scadenza` DATE)

BEGIN

declare exit handler for sqlexception, SQLSTATE "45001"

begin

rollback;

resignal;

end;

set transaction isolation level read committed;

start transaction;

INSERT INTO Auth VALUES(username, md5(password), 'user');

INSERT INTO Utenti VALUES(cf, nome, cognome, indirizzo, data\_nascita, citta\_nascita, username);

INSERT INTO CarteCredito VALUES (numero\_carta, cf, cvv, data\_scadenza);

commit;

END

**Operazione U01:**

Nella seguente operazione ho definito una transazione con livello di isolamento ‘read committed’

per evitare la ‘lettura sporca’, poiché nel momento in cui un utente A fa un’offerta (t1) e un utente B invece esegue l’operazione U02 (t2) e quest’ultima va in abort la transazione t1 potrebbe leggere il dato scritto da t2 che però non è mai stato salvato, in quanto t2 è andata in abort.

CREATE PROCEDURE `make\_offer`(IN `utente` CHAR(16), IN `importo` FLOAT UNSIGNED, IN `oggetto` CHAR(7))

BEGIN

DECLARE importo\_controfferta FLOAT UNSIGNED;

DECLARE controfferente CHAR(16);

DECLARE EXIT HANDLER FOR SQLEXCEPTION, SQLSTATE '45001', SQLSTATE '45002'

begin

rollback;

resignal;

end;

set transaction isolation level read commited;

start transaction;

SELECT Importo, Utente

FROM Controfferte

WHERE Oggetto=oggetto INTO importo\_controfferta, controfferente;

IF importo\_controfferta >= importo AND importo\_controfferta IS NOT NULL THEN

INSERT INTO Offerte VALUES (importo+0.50, oggetto, controfferente, NOW());

SIGNAL SQLSTATE '45002' set message\_text = "[OfferError] There is a counteroffer higher then your offer!";

ELSE

INSERT INTO Offerte VALUES(importo, oggetto, utente, NOW());

END IF;

commit;

END

**Operazione U02:**

CREATE PROCEDURE `set\_counteroffer`(IN `importo` FLOAT, IN `utente` CHAR(16), IN `oggetto` CHAR(7))

BEGIN

DECLARE EXIT HANDLER FOR SQLEXCEPTION, SQLSTATE "45001", SQLSTATE "45002"

BEGIN

ROLLBACK;

RESIGNAL;

END;

INSERT INTO Controfferte VALUES (importo, utente, oggetto);

END

**Operazione U03:**

CREATE PROCEDURE `user\_auctions\_informations`(IN `var\_utente` CHAR(16))

BEGIN

DECLARE EXIT HANDLER FOR SQLEXCEPTION

BEGIN

ROLLBACK;

RESIGNAL;

END;

SELECT `Codice`, `Descrizione`, ScadenzaAsta, NumeroOfferte, OffertaMassima

FROM Aste

WHERE `ScadenzaAsta` > CURRENT\_TIMESTAMP() AND Partecipante=var\_utente

ORDER BY(ScadenzaAsta);

END

**Operazione U04:**

CREATE PROCEDURE `auctions\_in\_progress\_informations`()

BEGIN

SELECT

`Codice`,

`Stato`,

`Lunghezza`,

`Larghezza`,

`Altezza`,

`Descrizione`,

`BaseAsta`,

`ScadenzaAsta`,

`PrimoLivello`,

`SecondoLivello`,

`TerzoLivello`,

`OffertaMassima`,

`NumeroOfferte`

FROM Aste

WHERE `ScadenzaAsta` > CURRENT\_TIMESTAMP()

ORDER BY(ScadenzaAsta);

END

**Operazione U05:**

CREATE PROCEDURE `closed\_auctions\_informations`()

BEGIN

SELECT

`Codice`,

`Stato`,

`Lunghezza`,

`Larghezza`,

`Altezza`,

`Descrizione`,

`BaseAsta`,

`ScadenzaAsta`,

`PrimoLivello`,

`SecondoLivello`,

`TerzoLivello`,

`OffertaMassima`,

`NumeroOfferte`

FROM Aste

WHERE ScadenzaAsta<=NOW();

END

**Operazione A01:**

CREATE PROCEDURE `call\_auction`(IN `stato` VARCHAR(32), IN `lunghezza` TINYINT UNSIGNED, IN `larghezza` TINYINT UNSIGNED, IN `altezza` TINYINT UNSIGNED, IN `descrizione` TEXT, IN `base\_dasta` FLOAT UNSIGNED, IN `durata\_asta` TINYINT, IN `primo\_livello` VARCHAR(32), IN `secondo\_livello` VARCHAR(32), IN `terzo\_livello` VARCHAR(32))

BEGIN

DECLARE EXIT HANDLER FOR SQLEXCEPTION, SQLSTATE "45001"

BEGIN

ROLLBACK;

RESIGNAL;

END;

IF durata\_asta > 7 OR durata\_asta < 1 THEN

SIGNAL SQLSTATE "45001" SET MESSAGE\_TEXT = "[InputError] The duration of the asta must be between 1 and 7 days";

END IF;

INSERT INTO Oggetti VALUES (LEFT (UUID(),7), stato, lunghezza, larghezza, altezza, descrizione, base\_dasta, DATE\_ADD(NOW(), INTERVAL durata\_asta DAY), 0, primo\_livello, secondo\_livello, terzo\_livello);

END

**Operazione A02:**

CREATE PROCEDURE `insert\_category`(IN `primo\_livello` VARCHAR(32), IN `secondo\_livello` VARCHAR(32), IN `terzo\_livello` VARCHAR(32))

BEGIN

DECLARE EXIT HANDLER FOR SQLEXCEPTION, SQLSTATE '45001'

BEGIN

ROLLBACK;

RESIGNAL;

END;

INSERT INTO Categorie(PrimoLivello, SecondoLivello, TerzoLivello) VALUES (primo\_livello, secondo\_livello, terzo\_livello);

END

1. Indicare con E le entità, con R le relazioni [↑](#footnote-ref-2)