

# EBUY

## DATI DI INTERESSE:

### 1. UTENTI

#### 1.1 USERNAME

#### 1.2 DATA DI REGISTRAZIONE

#### 1.3 SE SONO PRIVATI O AZIENDE

##### 1.3.1 DEI VENDITORI PROFESSIONALI INTERESSA:

###### 1.3.1.1 URL VETRINA WEB

### 2. POST

#### 2.1 TITOLO

#### 2.1 SE SI TRATTÀ DI UN'ASTA O DI UN "COMPRALO SUBITO"

##### 2.2.1 DELLE ASTE INTERESSA:

###### 2.2.1.1 PREZZO INIZIALE

###### 2.2.1.4 BIDS (REQ. 3)

###### 2.2.1.2 RIALZO

###### 2.2.1.5 DELLE ASTE CHIUSE INTERESSA:

###### 2.2.1.3 ISTANTE DI SCADENZA

###### 2.2.1.5.1 BID VINCITORE

##### 2.2.2 DEI "COMPRALO SUBITO" INTERESSA:

###### 2.2.2.1 PREZZO DI VENDITA

###### 2.2.2.2 EVENTUALE ASSURENTE (REQ. 1) QUOTENTE

###### 2.2.2.3 VENDITORE (REQ. 1)

###### EVENTUALE

###### 2.3 SE L'ARTICOLO È NUOVO O USATO

###### 2.3.1 DEGLI ARTICOLI NUOVI INTERESSA:

###### 2.3.1.1 DURATA DELLA GARANZIA (MIN. 2 ANNI)

###### 2.3.2 DEGLI ARTICOLI USATI INTERESSA:

###### 2.3.2.1 CONDIZIONI (OTTIMO BUONO, DISCRETO, DA SISTEMARE)

###### 2.3.2.2 GARANZIA (FACOLTATIVA)

###### 2.5 CATEGORIE (REQ. 4) 2.6 DATA E ORA PUBBL.

###### 2.7 PAGAMENTI ACCETTATI

###### 3. BIDS

###### 2.8 FEEDBACK (REQ. 5)

###### (CC O BONIFICO)

###### 3.1 UTENTE BIDDER (REQ. 1)

###### 3.2 ASTA DI RIFERIMENTO (REQ. 2.1.1)

###### 3.3 ISTANTE

### 4. CATEGORIE

#### 4.1 NOME

#### 4.2 EVENTUALI SOTTO-CATEGORIE (REQ. 4)

### 5. FEEDBACK

#### 5.1 UTENTE (REQ. 1)

#### 5.2 ARTICOLO DI RIFERIMENTO (REQ. 2.2.2)

#### 5.3 PUNTEGGIO (DA 0 A 5)

#### 5.4 EVENTUALE COMMENTO

## FUNZIONALITÀ RICHIESTE:

### 6. UN UTENTE REGISTRATO DEVE ESSERE IN GRADO DI METTERE IN VENDITA UN ARTICOLO TRAMITE POST.

### 7. GLI UTENTI PRIVATI DEVONO POTER PROPORRE BID SULLE ASTE ATTUALMENTE APERTE. IL PREZZO PROPOSTO SARÀ PARI ALLA PROPOSTA PRECEDENTE PIÙ IL RIALZO STABILITO DAL VENDITORE. AD ASTA CONCLUSA, L'UTENTE CHE HA EFFETTUATO L'ULTIMO BID SI AGGIUDICHERÀ L'ARTICOLO.

### 8. GLI UTENTI DEVONO ESSERE IN GRADO DI ACQUISTARE ARTICOLI MEDIANTE LA FORMULA "COMPRALO SUBITO". PER VALUTARE LA SERVITÀ NEL VENDITORE, L'UTENTE DOVRÀ POTER ESSERE IN grado di ACCORDARSI SULLA FORMA DI PAGAMENTO. IN QUESTO MODO L'UTENTE SOFFRE POTER ESTIMERE UN FEEDBACK SULLA VENDITA DI UN ARTICOLO DA UN VENDITORE MEDIANTE

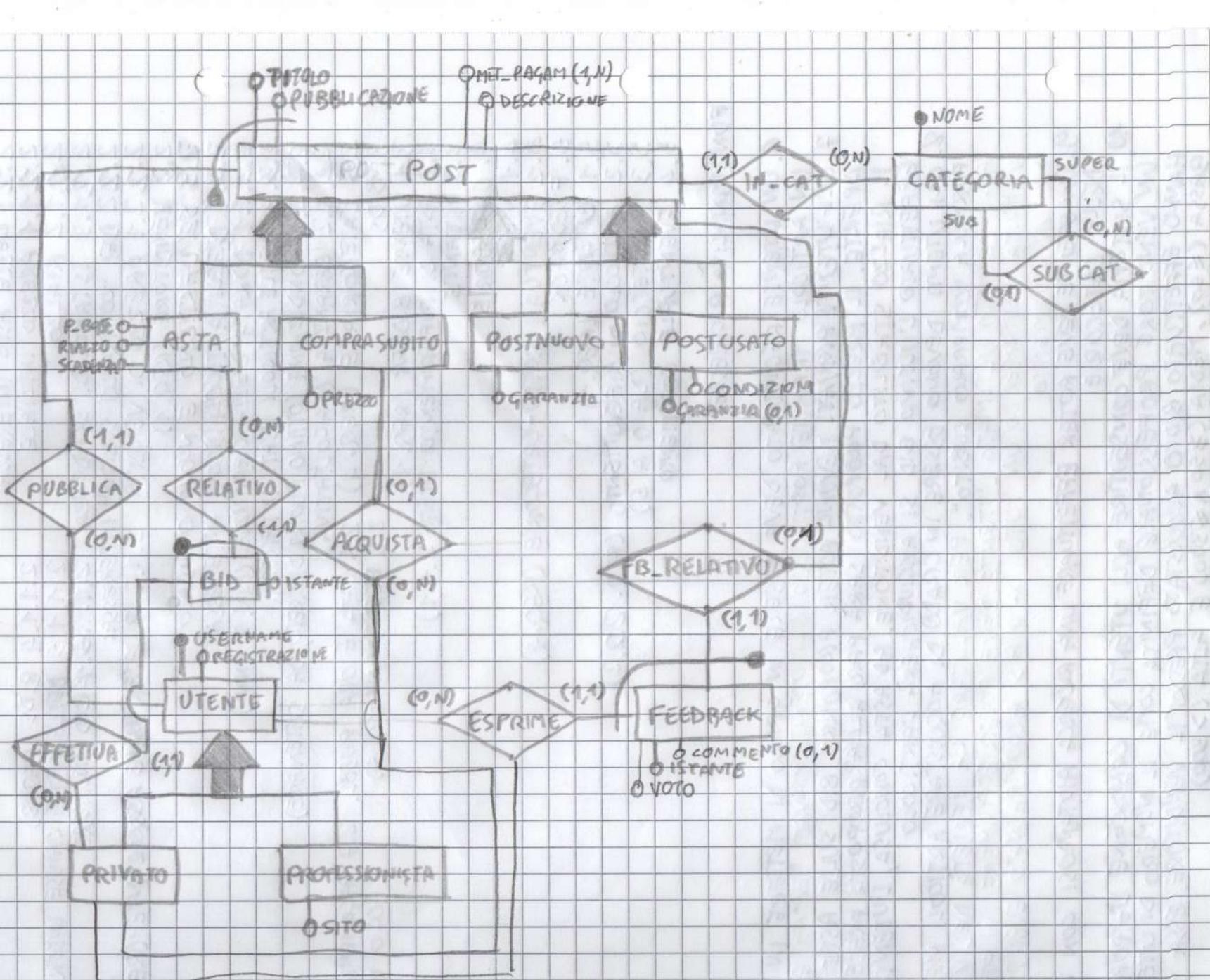
### 9. UN UTENTE DEVE POTER EFFETTUARE UNA RICERCA AIUTANDOSI CON IL SISTEMA NELLE CATEGORIE

### 10. IL SISTEMA DEVE CONSENTIRE AGLI UTENTI DI CONOSCERE LA POPOLARITÀ DI UN VENDITORE PROFESSIONALE DEFINITA COME IL NUMERO DI UTENTI CHE HANNO EFFETTUATO BID O ACQUISTATO PRESSO IL VENDITORE. ESSA SI CLASSIFICA BASSA (<50), MEDIA (350 A <300) E ALTA (>300)

11. GLI UTENTI CHE ACQUISTANO UN ARTICOLO DEVONO POTER ESPRIMERE UN FEEDBACK CARATTERIZZATO DA UN PUNTEGGIO DA 0 A 5, EVENTUALMENTE ACCOMPAGNATO DA UN COMMENTO.

12. PER OGNI UTENTE, IL SISTEMA DEVE PERMETTERE DI CALCOLARNE L'AFFIDABILITÀ, DEFINITA COME LA MEDIA ARITMETICA  $m$  DEI FEEDBACK, DIMINUITA DI UN FATTORE CHE CRESCE ALL'AUMENTARE DEI FEEDBACK NEGATIVI ( $\leq 2$ ). TECNICAMENTE, SIA  $Z$  LA FRAZIONE DEI FEEDBACK NEGATIVI, L'AFFIDABILITÀ È CALCOLATA COME:

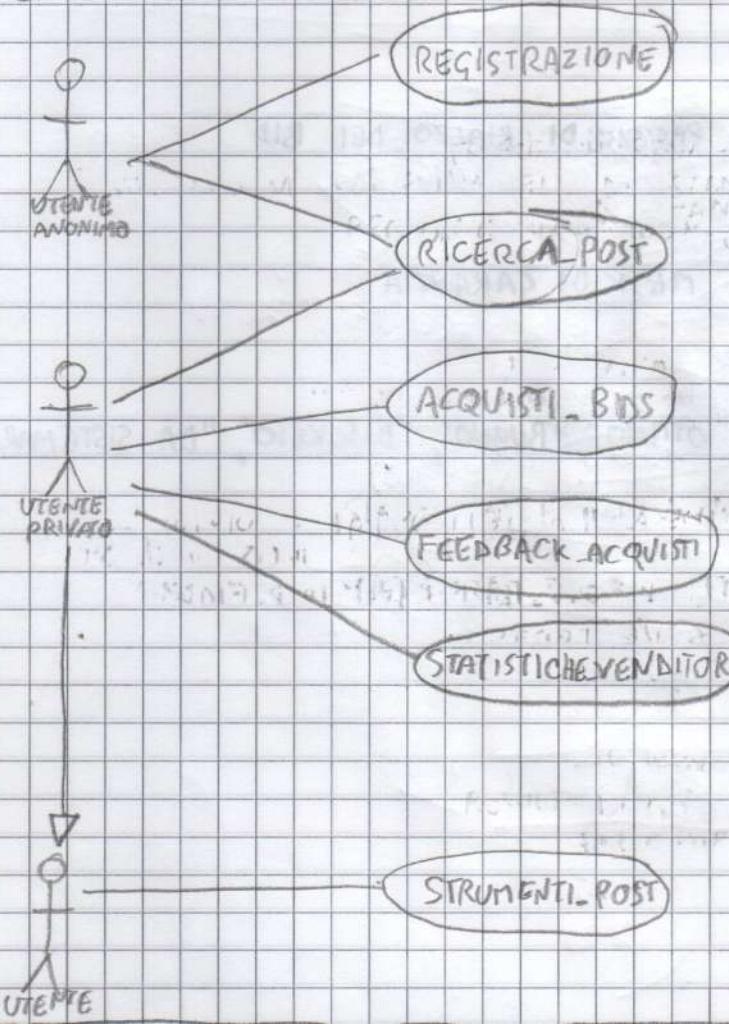
$$A = \frac{(1-z)m}{5}$$



# DIZIONARIO DEI DATI:

ATTRIBUTO	TIPO	NOTE
ENTITÀ POST:		
TITOLO	STRINGA	
DESCRIZIONE	STRINGA	
PUBBLICAZIONE	DATAORA	
MET_PAGAM (1..N)	{ CC, BO }	METODI DI PAG. ACCETTATI: CC E/O BONIFICO
ENTITÀ COMARA SUBITO:		
PREZZO	VALUTA > 0	
ENTITÀ ASTA:		
P_BASE	VALUTA > 0	
RIALZO	VALUTA > 0	PREZZO DI RIALZO DEL BID
SCADENZA:	DATAORA	
ENTITÀ POSTNUOVO:		
GARANZIA	INTERO >= 24	MESI DI GARANZIA
ENTITÀ POSTUSATO:		
GARAZIA (0..1)	INTERO > 0	" "
CONDIZIONI	{ O, B, D, SB }	"OTTIMO", "BUONO", "DISCRETO", "DA SISTEMARE"
ENTITÀ CATEGORIA:		
NOME	STRINGA	STR.
ENTITÀ BID:		
ISTANTE	DATAORA	
ENTITÀ UTENTE:		
USERNAME	STRINGA	
REGISTRAZIONE	DATAORA	
ENTITÀ PROFESSIONISTA:		
SITO	STR. URL	
ENTITÀ FEEDBACK:		
ISTANTE	DATAORA	
VOTO	INTERO [0, 5]	
COMMENTO (0..1)	STRINGA	

UML



## VINCOLI ESTERNI

V. POST. NO-FUTURO:  $\forall p, d [ \text{POST}(p) \wedge \text{PUBBLICAZIONE}(p, d) ] \rightarrow d \leq \text{ADESSO}$

V. CATEGORIA. NO-RIC:  $\forall c [ \text{CATEGORIA}(c) ] \rightarrow \neg \text{SUBCAT}(c, c)$

V. BID. NO-FUTURO:  $\forall b, i [ \text{BID}(b) \wedge \text{ISTANTE}(b, i) ] \rightarrow i \leq \text{ADESSO}$

V. UTENTE. NO-FUTURO:  $\forall u, r [ \text{UTENTE}(u) \wedge \text{REGISTRAZIONE}(u, r) ] \rightarrow r \leq \text{ADESSO}$

V. FEEDBACK. NO-FUTURO:  $\forall f, i [ \text{FEEDBACK}(f) \wedge \text{ISTANTE}(f, i) ] \rightarrow i \leq \text{ADESSO}$

V. FEEDBACK. TRANSAZIONE:  $\forall f, u, o [ \text{FEEDBACK}(f) \wedge \text{ESPRIME}(u, f) \wedge \text{RELATIVO}(f, o) ]$   
 $\rightarrow [ \text{COMPRASUBITO}(o) \wedge \text{ACQUISTA}(u, o) ] \vee [ \text{ASTA}(o) \wedge$   
 $(\exists b [ \text{BID}(b) \wedge \exists b' \exists t, t' [ \text{ISTANTE}(b, t) \wedge \text{ISTANTE}(b', t') \wedge t > t' ] )$   
 $\wedge \text{EFFETTUA}(u, b) ]$

## SPECIFICHE USE-CASE REGISTRAZIONE

REGISTRA(USERNAME: STRINGA, TIPO: {PRIV, PRO}, URL: STRINGA) : UTENTE

PRECONDIZIONI:  $\neg \exists u [UTENTE(u) \wedge USERNAME(u, \text{USERNAME})] \wedge (TIPO = \text{PRO}) \rightarrow \text{STRINGA}(\text{URL})$

POSTCONDIZIONI:

MODIFICA AL LIVELLO ESTENSIONALE DEI DATI:

NUOVI ELEMENTI DEL DOMINIO DI INTERPRETAZIONE: u

ELEMENTI DEL DOMINIO DI INTERPRETAZIONE CHE NON ESISTONO PIÙ: NESSUNO

NUOVE ENNUPLE DI PREDICATI:

UTENTE(u)

PRIVATO(u)  $\leftrightarrow$  TIPO = PRIV

PROFESSIONISTA(u)  $\leftrightarrow$  TIPO = PRO

SITO(u, URL)  $\leftrightarrow$  TIPO = PRO

USERNAME(u, USERNAME)

REGISTRAZIONE(u, ADERSO)

ENNUPLE DI PREDICATI CHE NON ESISTONO PIÙ: NESSUNA

VALORE DI RITORNO: RESULT = u

## SPECIFICHE USE-CASE RICERCA\_POST:

CERCA\_POST(QUERY: STRINGA, CATEGORIE: CATEGORIA(0, N)) : POST(0, N)

PRECONDIZIONI:  $\exists p [POST(p)] \wedge (|CATEGORIE| > 0) \rightarrow \exists c CATEGORIA(c)$

POSTCONDIZIONI:

MODIFICA AL LIVELLO ESTENSIONALE DEI DATI: NESSUNA

VALORE DI RITORNO: SIA P = { (p, t) | POST(p)  $\wedge$  TITOLO(p, t)  $\wedge$  QUERY  $\subseteq$  TITOLO  $\wedge$

$(\exists c \in \text{CATEGORIE} [IN\_CAT(p, c)]) \}$

RISULTATO SUB = { s | CATEGORIA(s)  $\wedge$  SUBCAT(s, i) }  $\forall i \in \text{CATEGORIE}$

SIA R =  $\bigcup_{i \in \text{CATEGORIE}} "RICERCA\_POST". \text{CERCA\_POST}(QUERY, SUB_i)$

RESULT =  $\bigcup \{ p \mid (p, t) \in P \wedge t \in R \}$

## SPECIFICHE USE-CASE ACQUISTI\_BIDS

COMPRALO\_SUBITO(u: PRIVATO, cs: COMPRASUBITO)

PRECONDIZIONI:  $\exists a [PRIVATO(a) \wedge ACQUISTA(a, cs)]$

POSTCONDIZIONI:

MODIFICA AL LIVELLO ESTENSIONALE DEI DATI:

DOMINIO DI INTERPRETAZIONE: INVARIATO

NUOVE ENNUPLE DI PREDICATI: ACQUISTA(u, cs)

ENNUPLE DI PREDICATI CHE NON ESISTONO PIÙ: NESSUNA

VALORI DI RITORNO: NESSUNO

PROPONI\_BID(u: PRIVATO, z: ASTA): VALUTA

PRECONDIZIONI:  $\exists s \text{ DATAORA}(s) \wedge \text{SCADENZA}(z, s) \wedge s > \text{ADESSO}$

POSTCONDIZIONI:

MODIFICA AL LIVELLO ESTENSIONALE DEI DATI:

NUOVI ELEMENTI DEL DOMINIO DI INTERPRETAZIONE: b

ELEMENTI DEL DOMINIO DI INTERPRETAZIONE CHE NON ESISTONO PIÙ: NESSUNO

NUOVE ENNUPLE DI PREDICATI:

BID(b)

ISTANTE(b, ADESSO)

EFFETTUAR(u, b)

RELATIVO(b, z)

ENNUPLE DI PREDICATI CHE NON VALGONO PIÙ: NESSUNA

VALORE DI RITORNO: SIA BASE | VALUTA(BASE)  $\wedge$  P\_BASE(z, BASE)

$$\text{OLD_BIDS} = \{b' \mid \text{BID}(b') \wedge \text{RELATIVO}(b', z)\}$$

$$R \mid \text{VALUTA}(R) \wedge \text{RIALZO}(z, R)$$

$$\text{RESULT} = \text{BASE} + (\text{OLD_BIDS} + 1) \cdot R$$

SPECIFICHE USE-CASE FEEDBACK\_ACQUISTI:

FEEDBACK\_ASTA(u: PRIVATO, z: ASTA, VOTO: INTERO[0, 5], COMMENTO: STRINGA(0, 1))

PRECONDIZIONI: "STRUMENTI\_POST". VINCITORE\_ASTA(z) = u

POSTCONDIZIONI:

MODIFICA AL LIVELLO ESTENSIONALE DEI DATI:

NUOVI ELEMENTI DEL DOMINIO DI INTERPRETAZIONE: f

ELEMENTI DEL DOMINIO DI INTERPRETAZIONE CHE NON ESISTONO PIÙ: NESSUNO

NUOVE ENNUPLE DI PREDICATI:

FEEDBACK(f)

VOTO(f, VOTO)

ISTANTE(f, ADESSO)

COMMENTO(f, COMMENTO)

ESPRIME(u, f)

FB\_RELATIVO(f, z)

ENNUPLE DI PREDICATI CHE NON VALGONO PIÙ: NESSUNA

VALORE DI RITORNO: NESSUNO

FEEDBACK\_CS(u: PRIVATO, cs: COMPRASURITO, VOTO: INTERO[0, 5], COMMENTO: STRINGA(0, 1))

PRECONDIZIONI: ACQUISTA(u, cs)

POSTCONDIZIONI:

MODIFICA AL LIVELLO ESTENSIONALE DEI DATI:

NUOVI ELEMENTI DEL DOMINIO DI INTERPRETAZIONE: f

ELEMENTI DEL DOMINIO DI INTERPRETAZIONE CHE NON ESISTONO PIÙ: NESSUNO

NUOVE ENNUPLE DI PREDICATI:

FEEDBACK(f)

VOTO(f, VOTO)

ISTANTE(f, ADESSO)

COMMENTO(f, COMMENTO)

ESPRIME(u, f)

FB\_RELATIVO(f, cs)

ENNUPLE DI PREDICATI CHE NON VALGONO PIÙ: NESSUNA

VALORE DI RITORNO: NESSUNO

**POST-ASTA-U**(TITOLO: STRINGA, DESCRIZIONE: STRINGA, BASE: VALUTA, RIALZO: VALUTA,  
 SCADENZA: DATAORA, MET\_PAGAM: {CC, BOB(1,N),  
 CAT: CATEGORIA, GARANZIA: INTERO > 0 (0,1),  
 CONDIZIONI: {O, B, D, S}, V: UTENTE) : ASTA

PRECONDIZIONI: SCADENZA > ADESSO

POSTCONDIZIONI:

MODIFICA AL LIVELLO ESTENSIONALE DEI DATI:

NUOVI ELEMENTI DEL DOMINIO DI INTERPRETAZIONE: 2

ELEMENTI DEL DOMINIO DI INTERPRETAZIONE CHE NON ESISTONO PIÙ: NESSUNO  
 NUOVE ENNUPLE DI PREDICATI:

POST(z)	CONDIZIONI(z, CONDIZIONI)
ASTA(z)	MET_PAGAM(z, MET_PAGAM)
POSTUSATO(z)	GARANZIA(z, GARANZIA)
TITOLO(z, TITOLO)	PUBBLICAZIONE(z, ADESSO)
DESCRIZIONE(z, DESCRIZIONE)	PUBBLICA(V, z)
P_BASE(z, BASE)	IN_CAT(z, CATEGORIA)
RIALZO(z, RIALZO)	

ENNUPLE DI PREDICATI CHE NON VALGONO PIÙ: NESSUNA

VALORE DI RITORNO: RESULT = 2

**POST-CS-N**(TITOLO: STRINGA, DESCRIZIONE: STRINGA, PREZZO: VALUTA, MET\_PAGAM: {CC, BOB(1,N),  
 CAT: CATEGORIA, GARANZIA: INTERO > 24, V: UTENTE}) : COMPRASUBITO

PRECONDIZIONI: NESSUNA

POSTCONDIZIONI:

MODIFICA AL LIVELLO ESTENSIONALE DEI DATI:

NUOVI ELEMENTI DEL DOMINIO DI INTERPRETAZIONE: CS

ELEMENTI DEL DOMINIO DI INTERPRETAZIONE CHE NON ESISTONO PIÙ: NESSUNO  
 NUOVE ENNUPLE DI PREDICATI:

POST(cs)	MET_PAGAM(cs, MET_PAGAM)
COMPRASUBITO(cs)	GARANZIA(cs, GARANZIA)
POSTNUOVO(cs)	PUBBLICAZIONE(cs, ADESSO)
TITOLO(cs, TITOLO)	PUBBLICA(V, cs)
DESCRIZIONE(cs, DESCRIZIONE)	IN_CAT(cs, CATEGORIA)
PREZZO(cs, PREZZO)	

ENNUPLE DI PREDICATI CHE NON VALGONO PIÙ: NESSUNA

VALORE DI RITORNO: RESULT = CS

**POST-CS-U**(TITOLO: STRINGA, DESCRIZIONE: STRINGA, PREZZO: VALUTA, MET\_PAGAM: {CC, BOB(1,N),  
 CAT: CATEGORIA, GARANZIA: INTERO > 0 (0,1), CONDIZ: {O, B, D, S},  
 V: UTENTE}) : COMPRASUBITO

PRECONDIZIONI: NESSUNA

POSTCONDIZIONI:

MODIFICA AL LIVELLO ESTENSIONALE DEI DATI:

NUOVI ELEMENTI DEL DOMINIO DI INTERPRETAZIONE: CS

ELEMENTI DEL DOMINIO DI INTERPRETAZIONE CHE NON ESISTONO PIÙ: NESSUNO

NUOVE ENNUPLE DI PREDICATI:

POST(cs)	CONDIZIONI(cs, CONDIZ)
COMPRASUBITO(cs)	MET_PAGAM(cs, MET_PAGAM)
POSTUSATO(cs)	GARANZIA(cs, GARANZIA)
TITOLO(cs, TITOLO)	PUBBLICAZIONE(cs, ADESSO)
DESCRIZIONE(cs, DESCRIZIONE)	PUBBLICA(V, cs)
PREZZO(cs, PREZZO)	IN_CAT(cs, CATEGORIA)

ENNUPLE DI PREDICATI CHE NON VALGONO PIÙ: NESSUNA

VALORE DI RITORNO: RESULT = CS

VINCITORE ASTA (a: ASTA); (PRIVATO, VALUTA)

PRECONDIZIONI:  $\exists b \text{ BID}(b) \wedge \text{RELATIVO}(b, a)$  ]  $\wedge$   $\exists s \text{ DATAORA}(s) \wedge \text{SCADENZA}(a, s) \wedge s < \text{ADESSO}$   
POSTCONDIZIONI:

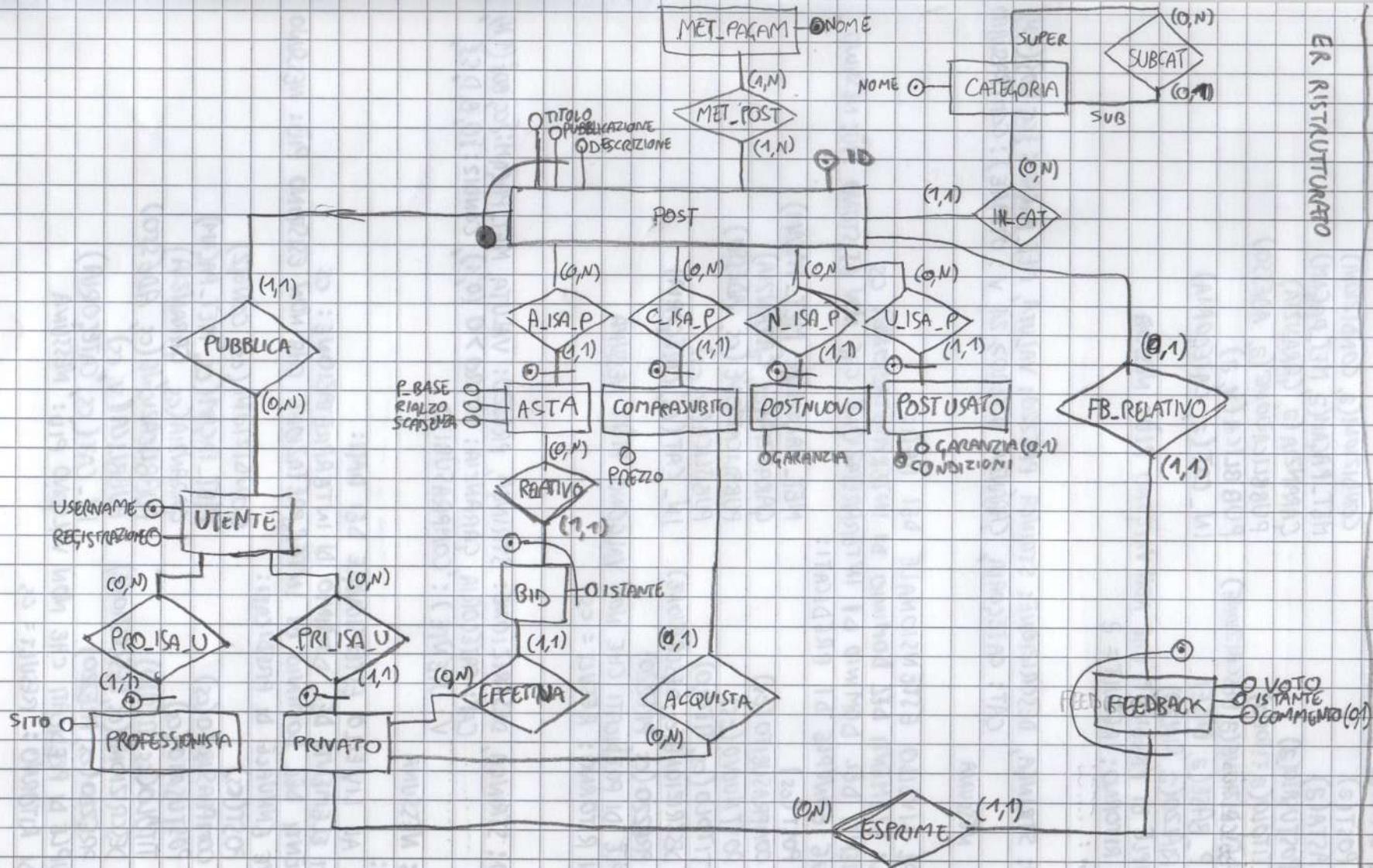
MODIFICA AL LIVELLO ESTENSIONALE DEI DATI: NESSUNA

VALORE DI RITORNO: SIA  $U = \{(u, b, i) \mid \text{PRIVATO}(u) \wedge \text{BID}(b) \wedge \text{EFFETTUATA}(u, b) \wedge$   
 $\text{ISTANTE}(b, i) \wedge \text{RELATIVO}(b, a)\}$

SIA  $\text{LAST} = \text{MAX}(\{i \mid \text{DATAORA}(i) \wedge \exists u, b \ (u, b, i) \in U\})$

SIA  $u \in (u, b, \text{LAST}) \in U$

RESULT = u



# DIZIONARIO DEI DATI (DBMS: POSTGRES):

## ENTITÀ POST:

TITOLO	VARCHAR(100)	ID: INT_POS (INTRODOTTO PER MOTIVI DI PERFORMANCE)
DESCRIZIONE	VARCHAR(300)	
PUBBLICAZIONE	DATETIME	

## ENTITÀ MET\_PAGAM:

NOME	VARCHAR(20)
------	-------------

## ENTITÀ COMPRASUBITO:

PREZZO	MONEY_POS
--------	-----------

## ENTITÀ ASTA:

P_BASE	MONEY_POS
RIALZO	MONEY_POS
SCADENZA	DATETIME

## ENTITÀ POSTNUOVO:

GARANZIA	INT_GE24
----------	----------

## ENTITÀ POSTUSATO:

GARANZIA(0,1)	INT_POS
CONDIZIONI	COND_T

## ENTITÀ CATEGORIA:

NOME	VARCHAR(50)
------	-------------

## ENTITÀ BID:

ISTANTE	DATETIME
---------	----------

## ENTITÀ UTENTE:

USERNAME	VARCHAR(50)
REGISTRAZIONE	DATE TIME

## ENTITÀ PROFESSIONISTA:

SITO	URL
------	-----

## ENTITÀ FEEDBACK

ISTANTE	DATETIME
VOTO	RATE_T
COMMENTO(0,1)	VARCHAR(1024)

## VINCOLI ESTERNI

V. POST. NO\_FUTURO:  $\forall p, d [ \text{POST}(p) \wedge \text{PUBBLICAZIONE}(p, d) ] \rightarrow d \leq \text{CURRENT\_TIMESTAMP}$

V. POST. DISG1:  $\forall p \text{ POST}(p) \rightarrow [\exists a A\text{-ISA-P}(a, p)] \rightarrow [\neg \exists c C\text{-ISA-P}(c, p)]$

V. POST. ISA1:  $\forall p \text{ POST}(p) \rightarrow \exists p' A\text{-ISA-P}(p', p) \vee C\text{-ISA-P}(p', p)$

V. POST. DISG2:  $\forall p \text{ POST}(p) \rightarrow [\exists n N\text{-ISA-P}(n, p)] \rightarrow [\neg \exists u U\text{-ISA-P}(u, p)]$

V. POST. ISA2:  $\forall p \text{ POST}(p) \rightarrow \exists p' N\text{-ISA-P}(p', p) \vee U\text{-ISA-P}(p', p)$

V. CATEGORIA. NO\_RIC:  $\forall c \text{ CATEGORIA}(c) \rightarrow \neg \text{SUBCAT}(c, c)$

V. BID. NO\_FUTURO:  $\forall b, i [ \text{BID}(b) \wedge \text{ISTANTE}(b, i) ] \rightarrow i \leq \text{CURRENT\_TIMESTAMP}$

V. UTENTE. NO\_FUTURO:  $\forall u, r [ \text{UTENTE}(u) \wedge \text{REGISTRATORE}(u, r) ] \rightarrow r \leq \text{CURRENT\_TIMESTAMP}$

V. UTENTE. DISG:  $\forall u \text{ UTENTE}(u) \rightarrow [\exists pr \text{ PRI-ISA-U}(pr, u)] \rightarrow [\neg \exists pf \text{ PRO-ISA-U}(pf, u)]$

V. UTENTE. ISA:  $\forall u \text{ UTENTE}(u) \rightarrow \exists u' \text{ PRI-ISA-U}(u', u) \vee \text{PRO-ISA-U}(u', u)$

V. FEEDBACK. NO\_FUTURO:  $\forall f, i [ \text{FEEDBACK}(f) \wedge \text{ISTANTE}(i) ] \rightarrow i \leq \text{CURRENT\_TIMESTAMP}$

V. FEEDBACK. TRANSAZIONE:  $\forall f, u, p [ \text{FEEDBACK}(f) \wedge \text{ESPRIME}(u, f) \wedge \text{RELATIVO}(f, p) ] \rightarrow [\exists ab \text{ ABASTA}(ab) \wedge A\text{-ISA-P}(ab, p) \wedge \text{RELATIVO}(ab, u) \wedge \text{EFFETTUATA}(u, ab) \wedge (\neg \exists b', t, t' \text{ ISTANTE}(b', t) \wedge \text{BID}(b') \wedge \text{ISTANTE}(b', t') \wedge t' > t)] \vee [\exists c \text{ COMPRASUBITO}(c) \wedge C\text{-ISA-P}(c, p) \wedge \text{ACQUISTA}(u, c)]$

## TIPI PERSONALIZZATI:

MONEY\_POS : MONEY CHECK VALUE > 0

INT\_GE29 : INTEGER CHECK VALUE  $\geq 29$

INT\_POS : INTEGER CHECK VALUE > 0

COND\_T : ENUM ('o', 'b', 'd', 's')

URL : VARCHAR(30) CHECK VALUE ~ 'http[s]://[2-20-9\_-.]+.[2-2]+'

RATE\_T : INTEGER CHECK VALUE BETWEEN 0 AND 5

## SCHEMA RELAZIONALE

POST (ID: INT, POS: UTENTE: VARCHAR(50), TITOLO: VARCHAR(100), PUBBLICAZIONE: DATETIME,  
DESCRIZIONE: VARCHAR(300), CATEGORIA: VARCHAR(50))  
VINCOLO CHIAVE: (UTENTE, TITOLO, PUBBLICAZIONE)  
|| FOREIGN KEY: UTENTE REFERENCES UTENTE(USERNAME)  
|| " " : CATEGORIA REFERENCES CATEGORIA(NOME)  
|| INCLUSIONE: ID ⊆ MET\_POST(POST)  
|| ENNUPLA: PUBBLICAZIONE ≤ CURRENT\_TIMESTAMP.

MET\_PAGAM (NOME: VARCHAR(20))  
VINCOLO INCLUSIONE: NOME ⊆ MET\_POST(MET)

MET\_POST (POST: INT, POS: MET: VARCHAR(20))  
VINCOLO FOREIGN KEY: POST REFERENCES POST(ID)  
|| " " : MET REFERENCES MET\_PAGAM(NOME)

CATEGORIA (NOME: VARCHAR(50))

SUBCAT (SUB: VARCHAR(50), SUPER: VARCHAR(50))  
VINCOLO FOREIGN KEY: SUB REFERENCES CATEGORIA(NOME)  
|| " " : SUPER || " " :  
|| ENNUPLA: SUB ≠ SUPER

ASTA (POST: INT, POS: P\_BASE: MONEY, RIALZO: MONEY, SCADENZA: DATETIME)  
VINCOLO FOREIGN KEY: POST REFERENCES POST(ID)  
|| ENNUPLA: SCADENZA > CURRENT\_TIMESTAMP

COMPRASUBITO (POST: INT, POS: PREZZO: MONEY)  
VINCOLO FOREIGN KEY: POST REFERENCES POST(ID)

POSTNUOVO (POST: INT, POS: GARANZIA: INT, GE24)  
VINCOLO FOREIGN KEY: POST REFERENCES POST(ID)

POSTUSATO (POST: INT, POS: GARANZIA\*: INT, CONDIZIONI: COND\_T)  
VINCOLO FOREIGN KEY: POST REFERENCES POST(ID)

UTENTE (USERNAME: VARCHAR(50), REGISTRAZIONE: DATETIME)  
VINCOLO ENNUPLA: REGISTRAZIONE ≤ CURRENT\_TIMESTAMP

PROFESSIONISTA (USERNAME: VARCHAR(50), SITO: URL)  
VINCOLO FOREIGN KEY: USERNAME REFERENCES UTENTE(USERNAME)

PRIVATO (USERNAME: VARCHAR(50))  
VINCOLO FOREIGN KEY: USERNAME REFERENCES UTENTE(USERNAME)

BID (UTENTE: VARCHAR(50), ISTANTE: DATETIME, ASTA: WT\_POS)  
VINCOLO FOREIGN KEY: UTENTE REFERENCES UTENTE(USERNAME)  
|| " " : ASTA REFERENCES ASTA(POST)  
|| ENNUPLA: ISTANTE ≤ CURRENT\_TIMESTAMP

ACQUISTA (CS: INT, POS: UTENTE: VARCHAR(50))  
VINCOLO FOREIGN KEY: CS REFERENCES COMPRASUBITO(POST)  
|| " " : UTENTE REFERENCES UTENTE(USERNAME)

FEEDBACK (UTENTE: VARCHAR(50), POST: INT, POS: VOTO: RATE\_T, ISTANTE: DATETIME, COMMENTO\*: VARCHAR(1000))  
VINCOLO FOREIGN KEY: UTENTE REFERENCES UTENTE(USERNAME)  
|| " " : POST REFERENCES POST(ID)  
|| ENNUPLA: ISTANTE ≤ CURRENT\_TIMESTAMP

## PROGETTAZIONE DEI VINCOLI ESTERNI:

### TRIGGER PER V. POST. DISG1:

#### - OPERAZIONI:

- INSERIMENTO O MODIFICA IN ASTA
- " " IN COMPRASUBITO

- ISTANTE DI INVOCAZIONE: PRIMA DELL'OPERAZIONE

#### - FUNZIONE:

SIA ERROR = FALSE

SE L'OPERAZIONE INSERISCE O MODIFICA IN ASTA:

SIA NEW L'ENNUPLA INSERITA O DOPO LA MODIFICA

ERROR := SELECT EXIST (SELECT 1 FROM COMPRASUBITO WHERE POST = NEW.POST)

SE L'OPERAZIONE INSERISCE O MODIFICA IN COMPRASUBITO:

SIA NEW L'ENNUPLA INSERITA O DOPO LA MODIFICA

ERROR := SELECT EXIST (SELECT 1 FROM ASTA WHERE POST = NEW.POST)

SE ERROR = TRUE BLOCCA L'OPERAZIONE

ALTRIMENTI CONSENTI L'OPERAZIONE

### TRIGGER PER V. POST. ISA1:

#### - OPERAZIONI: o modifica

- INSERIMENTO IN POST

- CANCELLAZIONE O MODIFICA IN ASTA

- " " IN COMPRASUBITO

- ISTANTE DI INVOCAZIONE: PRIMA DELL'OPERAZIONE

#### - FUNZIONE:

SIA ERROR = FALSE o modifica

SE L'OPERAZIONE INSERISCE IN POST:

SIA NEW L'ENNUPLA INSERITA

ERROR := SELECT EXIST ((SELECT 1 FROM ASTA WHERE POST = NEW.ID) UNION  
L'ATT. POST (SELECT 1 FROM COMPRASUBITO WHERE POST = NEW.ID))

SE L'OPERAZIONE MODIFICA O CANCELLA IN ASTA O COMPRASUBITO:

SIA OLD L'ENNUPLA ELIMINATA O PRIMA DELLA MODIFICA

ERROR := SELECT EXIST (SELECT 1 FROM POST WHERE ID = OLD.POST)

SE ERROR = TRUE BLOCCA L'OPERAZIONE

ALTRIMENTI CONSENTI L'OPERAZIONE

### TRIGGER PER V. POST. DISG2:

#### - OPERAZIONI:

- INSERIMENTO O MODIFICA IN POSTNUOVO

- " " IN POSTUSATO

- ISTANTE DI INVOCAZIONE: PRIMA DELL'OPERAZIONE

#### - FUNZIONE:

SIA ERROR = FALSE

SE L'OPERAZIONE INSERISCE O MODIFICA IN POSTNUOVO:

SIA NEW L'ENNUPLA INSERITA O DOPO LA MODIFICA

ERROR := SELECT EXIST (SELECT 1 FROM POSTUSATO WHERE POST = NEW.POST)

SE L'OPERAZIONE INSERISCE O MODIFICA IN POSTUSATO:

SIA NEW L'ENNUPLA INSERITA O DOPO LA MODIFICA

ERROR := SELECT EXIST (SELECT 1 FROM POSTNUOVO WHERE POST = OLD.POST)

SE ERROR = TRUE BLOCCA L'OPERAZIONE

ALTRIMENTI CONSENTI L'OPERAZIONE

## TRIGGER PER V. POST. ISA2:

### - OPERAZIONI:

- INSERIMENTO O MODIFICA IN POST
- MODIFICA O CANCELLAZIONE IN POSTNUOVO
- " " IN POSTUSATO
- ISTANTE DI INVOCAZIONE: PRIMA DELL'OPERAZIONE
- FUNZIONE:

SIA ERROR = FALSE

SE L'OPERAZIONE INSERISCE O MODIFICA IN POST:

SIA NEW L'ENNUPLA INSERITA O DOPO LA MODIFICA

ERROR := EXIST((SELECT 1 FROM POSTNUOVO WHERE POST=NEW.POST) UNION  
(SELECT 1 FROM POSTUSATO WHERE POST=NEW.POST))

SE L'OPERAZIONE MODIFICA L'ATTRIBUTO POST O CANCELLA IN POSTNUOVO O POSTUSATO:

SIA OLD L'ENNUPLA ELIMINATA O PRIMA DELLA MODIFICA

ERROR := EXIST (SELECT 1 FROM POST WHERE ID=OLD.POST)

SE ERROR = TRUE BLOCCA L'OPERAZIONE

ALTRIMENTI CONSENTI L'OPERAZIONE

## TRIGGER PER V. UTENTE. DISG:

### - OPERAZIONI:

- INSERIMENTO O MODIFICA IN PRIVATO
- " " IN PROFESSIONISTA
- ISTANTE DI INVOCAZIONE: PRIMA DELL'OPERAZIONE
- FUNZIONE:

SIA ERROR = FALSE

SE L'OPERAZIONE INSERISCE O MODIFICA IN PRIVATO:

SIA NEW L'ENNUPLA INSERITA O DOPO LA MODIFICA

ERROR := EXIST (SELECT 1 FROM PROFESSIONISTA WHERE USERNAME=NEW.USERNAME)

SE L'OPERAZIONE INSERISCE O MODIFICA IN PROFESSIONISTA:

ERROR := EXIST (SELECT 1 FROM PRIVATO WHERE USERNAME=NEW.USERNAME)

SE ERROR = TRUE BLOCCA L'OPERAZIONE

ALTRIMENTI CONSENTI L'OPERAZIONE

## TRIGGER PER V. UTENTE. ISA:

### - OPERAZIONI:

- INSERIMENTO O MODIFICA IN UTENTE
- MODIFICA O CANCELLAZIONE IN PRIVATO
- " " IN PROFESSIONISTA
- ISTANTE DI INVOCAZIONE: PRIMA DELL'OPERAZIONE
- FUNZIONE:

SIA ERROR = FALSE

SE L'OPERAZIONE INSERISCE IN UTENTE O MODIFICA USERNAME:

SIA NEW L'ENNUPLA INSERITA O DOPO LA MODIFICA

ERROR := EXIST((SELECT 1 FROM PRIVATO WHERE USERNAME=NEW.USERNAME)  
UNION (SELECT 1 FROM PROFESSIONISTA WHERE ))

SE L'OPERAZIONE MODIFICA O CANCELLA IN PRIVATO O PROFESSIONISTA:

ERROR := EXIST (SELECT 1 FROM UTENTE WHERE USERNAME=OLD.USERNAME)

SE ERROR = TRUE BLOCCA L'OPERAZIONE

ALTRIMENTI CONSENTI L'OPERAZIONE

## TRIGGER PER V. FEEDBACK. TRANSAZIONE:

- OPERAZIONI:

- INSERIMENTO O MODIFICA IN FEEDBACK

- ISTANTE DI INVOCAZIONE: PRIMA DELL'OPERAZIONE

- FUNZIONE:

- SIA ERROR = FALSE

- ERROR := NOT EXIST((SELECT 1 FROM ASTA WHERE POST = NEW.POST AND ASTA.POST =  
STRUMENTI\_POST.VINCITORE\_ASTA(POST) = NEW.UTENTE) UNION  
(SELECT 1 FROM COMPRASUBITO.C\_ACQUISTA\_A WHERE C.POST = A.CS  
AND A.UTENTE = NEW.UTENTE AND C.POST = NEW.POST))

SE ERROR = TRUE BLOCCA L'OPERAZIONE  
ALTRIMENTI CONSEGNA L'OPERAZIONE

## SPECIFICHE IMPLEMENTATIVE DEGLI USE-CASE:

### USE-CASE REGISTRAZIONE:

REGISTRAL\_PRIV (USERNAME: VARCHAR(50) : VARCHAR(50))

ERROR ← SELECT EXIST (SELECT 1 FROM UTENTE WHERE USERNAME = :USERNAME);

IF ERROR RAPPR. UN ERRORE INOLTRALO;

ELSE IF ERROR = TRUE THEN INOLTRA L'ERRORE 'UTENTE GIÀ ESISTENTE';

ELSE

OK ← BEGIN TRANSACTION;

INSERT INTO UTENTE VALUES (:USERNAME, CURRENT\_TIMESTAMP);

INSERT INTO PRIVATO VALUES (:USERNAME);

COMMIT WORK;

IF OK RAPPR. UN ERRORE THEN INOLTRALO;

ELSE RETURN :USERNAME;

REGISTRA\_PRO (USERNAME: VARCHAR(50), SITO: URL) : VARCHAR(50)

ERROR ← SELECT EXIST (SELECT 1 FROM UTENTE WHERE USERNAME = :USERNAME);

IF ERROR RAPPR. UN ERRORE THEN INOLTRALO;

ELSE IF ERROR = TRUE THEN INOLTRA L'ERRORE 'UTENTE GIÀ ESISTENTE';

ELSE

OK ← BEGIN TRANSACTION;

INSERT INTO UTENTE VALUES (:USERNAME, CURRENT\_TIMESTAMP);

INSERT INTO PROFESSIONISTA VALUES (:USERNAME, :SITO);

COMMIT WORK;

IF OK RAPPR. UN ERRORE THEN INOLTRALO;

ELSE RETURN :USERNAME;

### USE-CASE RICERCA\_POST:

CERCA\_POST(Q: VARCHAR(200)(0,M), C: VARCHAR(50)(0,N)) : INT\_POS(0,N)

RES ← ~~WORK~~ IN Q: DO

TIT\_DES ← SELECT CONCAT(TITOLO, ' ', DESCRIZIONE) AS TEXT, ID FROM POST POST  
WHERE CATEGORIA IN :C;

FOREACH WORD IN Q: DO

FOREACH ROW IN TIT\_DES: DO

IF WORD IN ROW.TEXT THEN RES ← RES U {ROW.ID};

DONE

NONE

FOREACH CAT IN :C DO

SUB ← SELECT SUB FROM SUBCAT WHERE SUPER = :CAT;

RES ← RES U RICERCA\_POST.CERCA\_POST(Q, SUB);

DONE

RETURN RES;

## USE-CASE ACQUISTI\_BIDS:

COMPRALO\_SUBITO (u: VARCHAR(50), cs: INT\_POS)

INS ← INSERT INTO ACQUISTA VALUES (:u,:cs);

IF INS RAPPR. UN ERRORE THEN INOLTRALO;

PROPONI\_BID (u: VARCHAR(50), z: INT\_POS): MONEY\_POS

sc ← SELECT SCADENZA FROM ASTA WHERE POST = :z;

IF SC RAPPR. ERRORE THEN INOLTRALO;

ELSE IF SC < CURRENT\_TIMESTAMP THEN INOLTRA ERR. "ASTA CHIUSA";

ELSE

INS ← INSERT INTO BID VALUES (:u, CURRENT\_TIMESTAMP, :z);

IF INS RAPPR. ERRORE THEN INOLTRALO;

ELSE

BASE ← SELECT P\_BASE FROM ASTA WHERE POST = :z;

IF BASE RAPPR. ERR. INOLTRALO;

R ← SELECT RIALZO FROM ASTA WHERE POST = :z; ↗ IF R ERR. INOLTRALO;

N ← SELECT COUNT(\*) FROM BIDS WHERE ASTA = :z;

IF N RAPPR. GRDRE THEN INOLTRALO;

ELSE RETURN BASE + R\*N;

## USE-CASE FEEDBACK\_ACQUISTI:

FEEDBACK (u: VARCHAR(50), p: INT\_POS, VOTO: RATE\_T COMM: VARCHAR(1024))

INS ← INSERT INTO FEEDBACK VALUES (:u, :p, :VOTO, CURRENT\_TIMESTAMP COMM);

// I TRIGGER VERIFicheranno che u ABbia EFFETTIVAMENTE ACQUISTATO p

IF INS RAPPR. ERRORE THEN INOLTRALO;

## USE-CASE STATISTICHE VENDITORI:

POPOLARITA (u: VARCHAR(50)): ENUM ('A' 'M' 'B')

OK ← SELECT EXIST(SELECT 1 FROM PROFESSIONISTA WHERE USERNAME = :u);

IF OK RAPPR. ERRORE THEN INOLTRALO;

ELSE IF OK = FALSE THEN INOLTRA ERR. "VENDITORE NON TROVATO";

ELSE

C ← SELECT COUNT(\*) FROM COMPRASUBITO cs ACQUISTA a, POST p WHERE  
cs.POST = a.POST AND cs.ID = p.ID AND p.UTENTE = :u;

IF C RAPPR. ERRORE THEN INOLTRALO;

B ← SELECT COUNT(\*) FROM BID b, ASTA a, POST p WHERE p.POST AND

a.POST = p.ID AND p.UTENTE = :u AND b.ASTA = a.POST;

IF B RAPPR. ERRORE THEN INOLTRALO;

ELSE RET:

POP ← B+C;

IF POP > 300 THEN RETURN 'A';

ELSE IF POP > 50 THEN RETURN 'M';

ELSE RETURN 'B';

APPIDABILITÀ (U: VARCHAR(50)) : REAL

OK ← SELECT EXIST (SELECT 1 FROM PROFESSIONISTA WHERE USERNAME = :U);  
IF OK RAPPR. ERRORE THEN INOLTRALO;  
ELSE  
F ← SELECT VOTO FROM FEEDBACK F, POST P WHERE F.POST = P.ID AND  
P.UTENTE = :U;

IF F RAPPRENDERRE THEN INOLTRALO;  
ELSE

IF |F| = 0 THEN m ← 0;  
ELSE m ← SUM(F) / |F|;  
z ← |F| / |{F | f ≤ z}|  
RETURN ((1 - z) \* m) / 5;

USE-CASE STRUMENTI\_POST:

POST\_ASTA\_N(TITOLO: VARCHAR(100), DESC: VARCHAR(300), BASE: MONEY\_POS, RIALZO: MONEY\_POS,  
SCAD: DATETIME, MP: VARCHAR(20)(1,N), CAT: VARCHAR(50), GAR: INT\_GE24,  
V: VARCHAR(50)): INT\_POS

ID ← PROSSIMO ID DISPONIBILE;

INS ← BEGIN TRANSACTION;

INSERT INTO POST VALUES (:ID, :V, :TITOLO, CURRENT\_TIMESTAMP, :DESC, :CAT);

INSERT INTO POSTNUOVO VALUES (:ID, :GAR);

INSERT INTO ASTA VALUES (:ID, :BASE, :RIALZO, :SCAD);

FOREACH m IN ARRAY :MP LOOP

INSERT INTO MET\_POST VALUES (:ID, m);

END LOOP.

COMMIT WORK;

IF INS RAPPR. ERRORE THEN INOLTRALO;

ELSE RETURN ID;

POST\_ASTA\_U(TITOLO: VARCHAR(100), DESC: VARCHAR(300), BASE: MONEY\_POS, RIALZO: MONEY\_POS,  
SCAD: DATETIME, MP: VARCHAR(20)(1,N), CAT: VARCHAR(50), GAR: INT\_POS,  
COND: COND\_T, V: VARCHAR(50)): INT\_POS

ID ← PROSSIMO ID DISPONIBILE;

INS ← BEGIN TRANSACTION;

INSERT INTO POST VALUES (:ID, :V, :TITOLO, CURRENT\_TIMESTAMP, :DESC, :CAT);

INSERT INTO POSTUSATO VALUES (:ID, :GAR, :COND);

INSERT INTO ASTA VALUES (:ID, :BASE, :RIALZO, :SCAD);

FOREACH m IN ARRAY :MP LOOP

INSERT INTO MET\_POST VALUES (:ID, m);

END LOOP.

COMMIT WORK;

IF INS RAPPR. ERRORE THEN INOLTRALO;

ELSE RETURN ID;

POST\_CS\_N(TITOLO: VARCHAR(100), DESC: VARCHAR(300), P: MONEY\_POS, MP: VARCHAR(20)(1,N),  
CAT: VARCHAR(50), GAR: INT\_GE24, V: VARCHAR(50)): INT\_POS

ID ← PROSSIMO ID DISPONIBILE;

INS ← BEGIN TRANSACTION;

INSERT INTO POST VALUES (:ID, :V, :TITOLO, CURRENT\_TIMESTAMP, :DESC, :CAT);

INSERT INTO POSTNUOVO VALUES (:ID, :GAR);

INSERT INTO COMPARASUGITO VALUES (:ID, :P);

FOREACH m IN ARRAY :MP LOOP

INSERT INTO MET\_POST VALUES (:ID, m);

END LOOP.

COMMIT WORK;

IF INS RAPPR. ERRORE THEN INOLTRALO;

ELSE RETURN ID;

POST\_CS\_U(TITOLO: VARCHAR(100), DESC: VARCHAR(300), P: MONEY, POS\_MP: VARCHAR(20)(1,N),  
CAT: VARCHAR(50), GAR\*: INT\_POS, COND: COND\_T, V: VARCHAR(50)): INT\_POS  
ID ← PROSSIMO ID DISPONIBILE;  
INS ← BEGIN TRANSACTION;  
INSERT INTO POST VALUES (:ID, :V.TITOLO, CURRENT\_TIMESTAMP, :DESC, :CAT);  
INSERT INTO POSTUSATO VALUES (:ID, :GAR, :COND);  
INSERT INTO COMPRASUBITO VALUES (:ID, :P);  
FOREACH m IN ARRAY :MP LOOP  
    INSERT INTO MET\_POST(:ID, m);  
END LOOP;  
COMMIT WORK;  
IF INS RAPPRE. ERRORE THEN INOLTRALO;  
ELSE RETURN ID;

VINCITORE ASTA (a: INT\_POS): VARCHAR(50)  
SCAD ← SELECT SCADENZA FROM ASTA WHERE POST = :a;  
IF SCAD RAPPRE. ERRORE THEN INOLTRALO;  
ELSE IF SCAD > CURRENT\_TIMESTAMP THEN  
    INOLTRA ERR. 'ASTA ANCORA APERTA';  
ELSE  
    B ← SELECT UTENTE FROM BID WHERE ASTA = :a ORDER BY ISTANTE DESC;  
    IF B RAPPRE. ERRORE THEN INOLTRALO;  
    ELSE RETURN B[0];