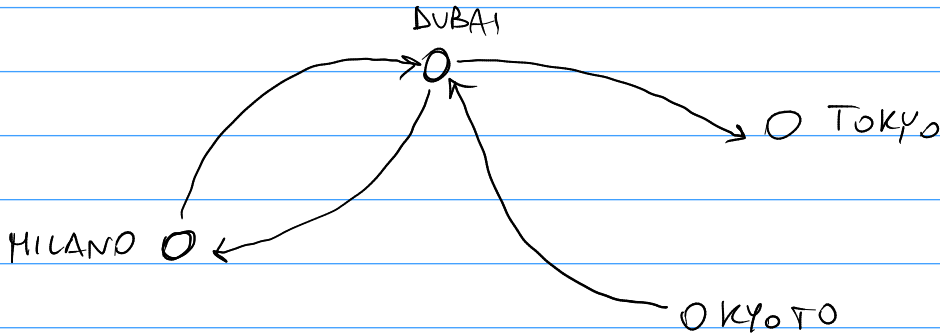


TUTORATO 20/05

AEROPORTO (Id, Città, Nazione, NumPiste)

VOLO (NumeroVolo, GiornoSett, IdAeropPart, IdAeropArrivo, Compagnia, OraPartenza, OraArrivo, CodiceAereo)

AEREO (Codice, Tipo, NumPasseggeri)



1. Cerchiamo di calcolare la capacità di trasporto dei singoli link.

```
CREATE VIEW TRASPORTO (Da, A, Passeggeri) AS
SELECT A1.Città, A2.Città, SUM(NumPasseggeri)
FROM VOLO JOIN AEREO ON CodAereo = Codice
JOIN AEROPORTO A1 ON IdAeropPart. = A1.Id
JOIN AEROPORTO A2 ON IdAeropArrivo = A2.Id
GROUP BY A1.Città, A2.Città
```

A1.C	A2.C	...	MP
A	B		
	C		

} x
} y

A	B	X
A	C	X
B	A	
B	C	

A	B	Σ
A	C	10
B	A	1
B	C	3
B	D	10

2. Calcolare la capacità per ogni coppia di città tenendo conto del tipo di link (simmetrico/asimmetrico)

link simmetrici

```
CREATE VIEW TOTALI (CUno, CDue, Passeggeri) AS  
(SELECT T1.Da, T1.A, T1.Passeggeri + T2.Passeggeri  
FROM TRASPORTO T1 JOIN TRASPORTO T2 ON  
T1.Da = T2.A AND T1.A = T2.Da
```

UNION

link asimmetrici

```
SELECT Da, A, Passeggeri  
FROM TRASPORTO  
WHERE (Da, A) NOT IN (SELECT A, Da  
FROM TRASPORTO)
```

link per i quali è possibile partire,
ma per tornare è necessario almeno uno
scalo.

3. Estrarre la coppia con la massima capacità

distinct → rimuovere i duplicati in caso di
coppia simmetrica con capacità massima.

```
SELECT DISTINCT CUno, CDue
```

```
FROM TOTALI  
WHERE Passeggeri = (SELECT MAX(Passeggeri)  
FROM TOTALI)
```

MAX dei P.

2. Estrarre in SQL le compagnie che servono la città di Bergamo con più di 70 voli alla settimana, usando aerei di un solo tipo per tutti i loro voli

1. Cerchiamo tutte le compagnie che partono da Bergamo

```
CREATE VIEW BGP (Compagnia, TipoAereo) AS
SELECT Compagnia, Tipo
FROM VOLO JOIN AEROPORTO A ON
    IdAerspPart = A.Id JOIN AEREO AE ON
    AE.Codia = CodAereo
WHERE A.Citta' = 'Bergamo'
```

```
CREATE VIEW BGA (Compagnia, TipoAereo) AS
SELECT Compagnia, Tipo
FROM VOLO JOIN AEROPORTO A ON
    IdAerspArrivo = A.Id JOIN AEREO AE ON
    AE.Codia = CodAereo
WHERE A.Citta = 'Bergamo'
```

```
2. CREATE VIEW BGTOT (Compagnia, TipoAereo) AS
    (SELECT * FROM BGP
     UNION ALL
     SELECT * FROM BGA)
```

→ conserva i duplicati

3. Ricerchiamo le compagnie che servono Bergamo usando un solo tipo di aereo.

```
CREATE VIEW BGUNO(Nome) AS
SELECT Compagnia
FROM BGTOT A
WHERE NOT EXISTS (SELECT * FROM BGTOT B
WHERE A.Compagnia = B.Compagnia
AND A.TipoAereo <> B.TipoAereo)
```

4. Selezioniamo tutte le compagnie che offrono più di 70 voli con un solo aereo.

```
SELECT Compagnia
FROM BGTOT
WHERE Compagnia IN (SELECT *
FROM BGUNO)
GROUP BY Compagnia
HAVING count(*) > 70 → più di 70 voli a settimana
```

PROGETTAZIONE CONCETTUALE

1. Comprendere i requisiti e le specifiche
2. Costruire dei modelli (ER)
3. Tradurre i modelli in schemi relazionali

2 Fasi:

1. COSTRUZIONE DI SCHEMI CONCETTUALI (E.R.)
2. COSTRUZIONE DI SCHEMI RELAZIONALI

TIPO	Tipi di Progettazione	
	<u>DIP. DA MODELLO</u>	<u>DIP. DA DBMS</u>
CONCETTUALE	NO	NO
LOGICA	SI	NO
FISICA	SI	SI

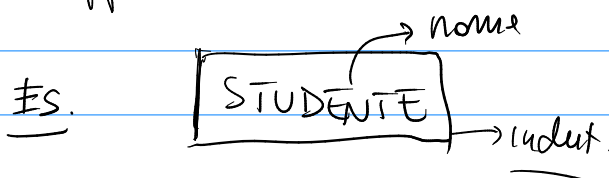
PROG. CONCETTUALE

- ① CLASSIFICARE: capacità di definire classi e oggetti del mondo reale
- ② AGGREGARE: Costituire classi complesse raggruppandone alcune più semplici.
- ③ GENERALIZZARE: Stabilire dei legami di sottoinsieme tra classi.

- Entità
- Associazioni
- Attributi

ENTITÀ

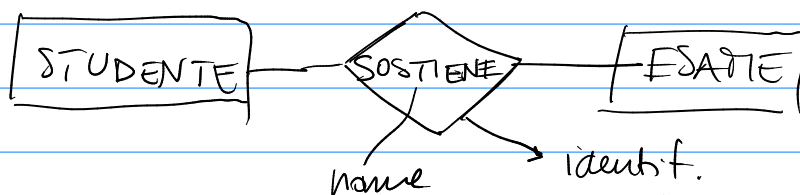
Rappresenta una classe di oggetti che è rilevante per l'applicazione in esame



ASSOCIAZIONI

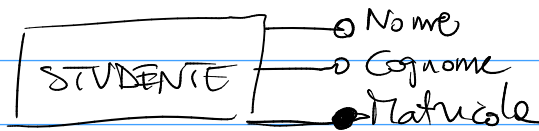
Rappresenta un'aggregazione di entità di interesse per l'applicazione

Es

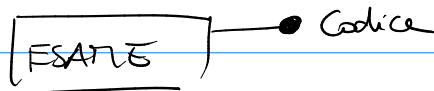


ATTRIBUTO

Rappresenta una caratteristica di un'entità o di una associazione, che è di interesse per l'applicazione. Di solito è esprimibile con un dominio elementare.

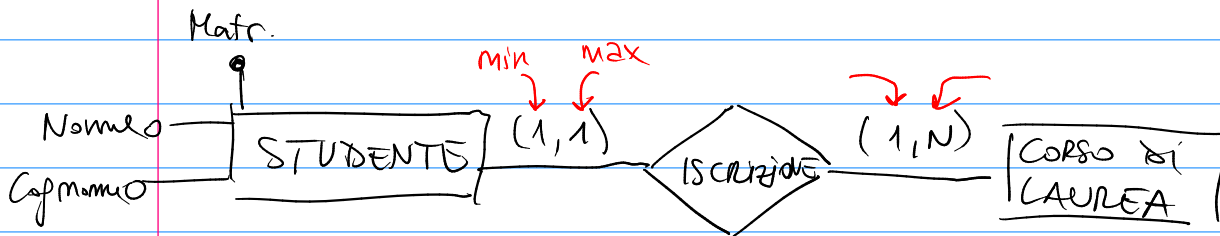


IDENTIFICATORI



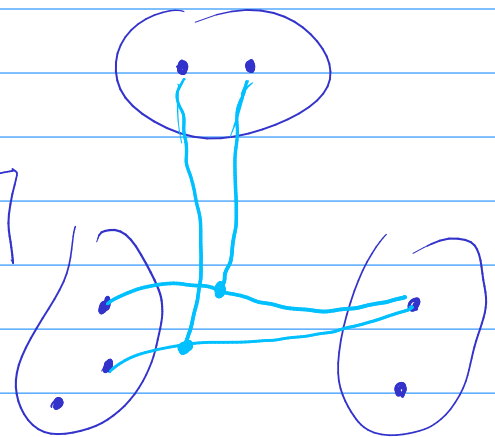
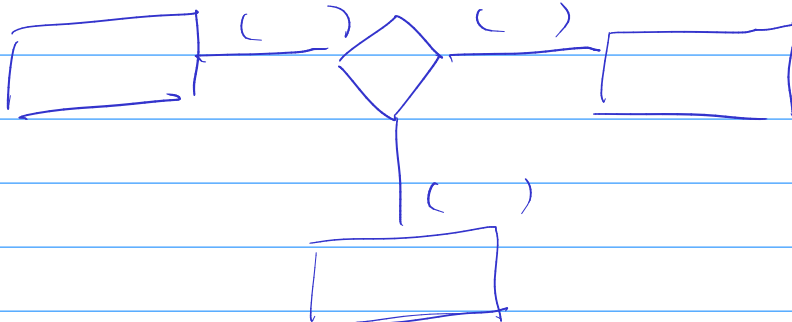
CARDINALITÀ DELLE ASSOCIAZIONI

Definisce il tipo di partecipazione di istanze di entità in legami di associazione.



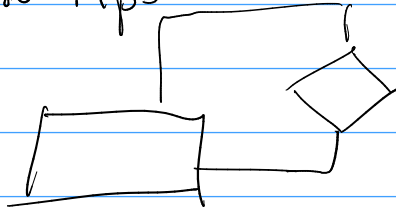
- (1,1)
- (1,n)
- (n,m)

ASSOCIAZIONE TERNAria



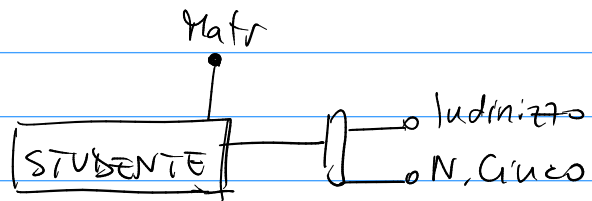
AUTO ASSOCIAZIONI

Sono possibili anche associazioni tra entità dello stesso tipo:

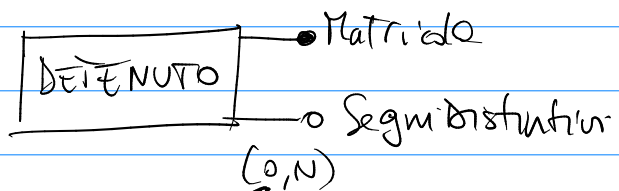


ATTRIBUTI

Ⓐ Composito

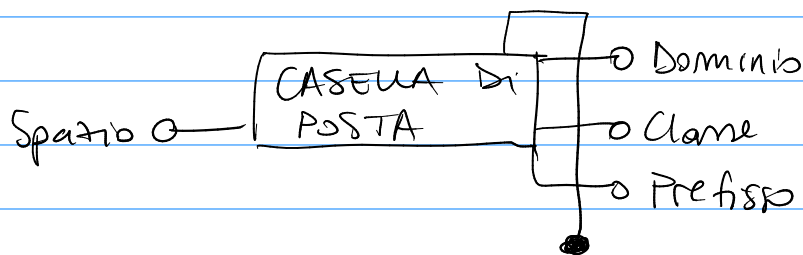


Ⓑ Multiplo

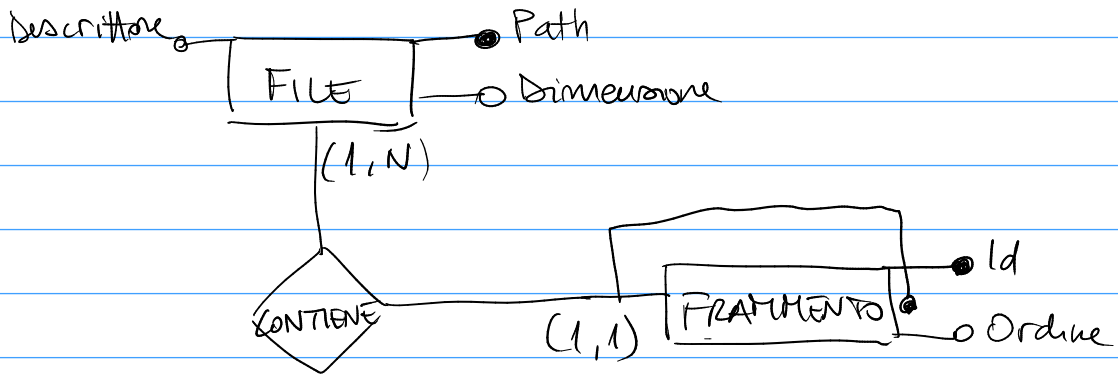


Ⓒ Opzionale

Si possono avere delle chiavi composte

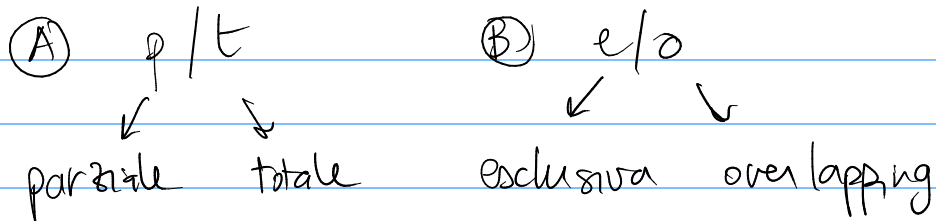
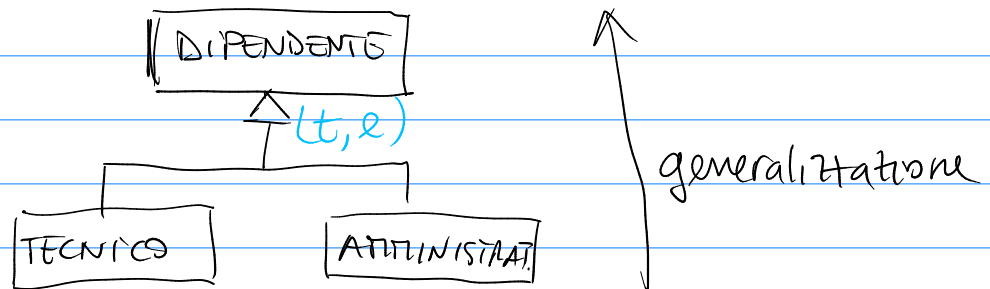
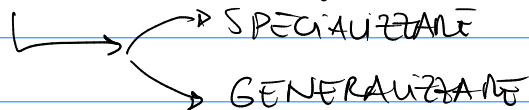


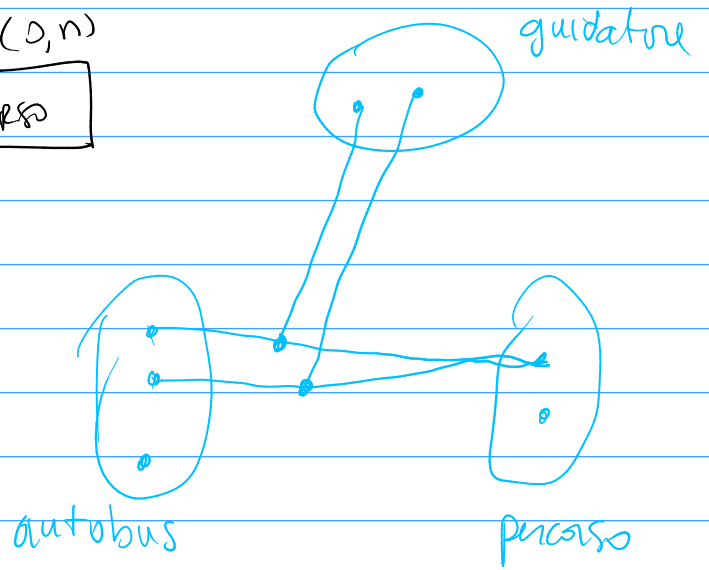
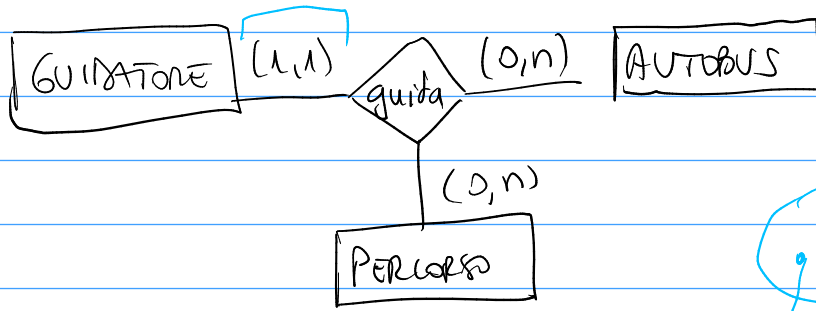
ENTITA' DEBOLI



GERARCHIE

Descrivono il legame logico tra entita' padre e figlio





Es - Tradurre in diagramma ER e schema logico

Una società vuole realizzare un database che supporti un sito web per aste online.

I clienti che si registrano al sito indicano un **username**, una **password** e **diversi dettagli del proprio conto bancario**.

Ciascun cliente può agire come **compratore** e **venditore**: nel ruolo di **venditore** il cliente può avviare molteplici **gare d'asta**, mentre nel ruolo di **compratore** il cliente può **effettuare delle offerte per diverse aste**.

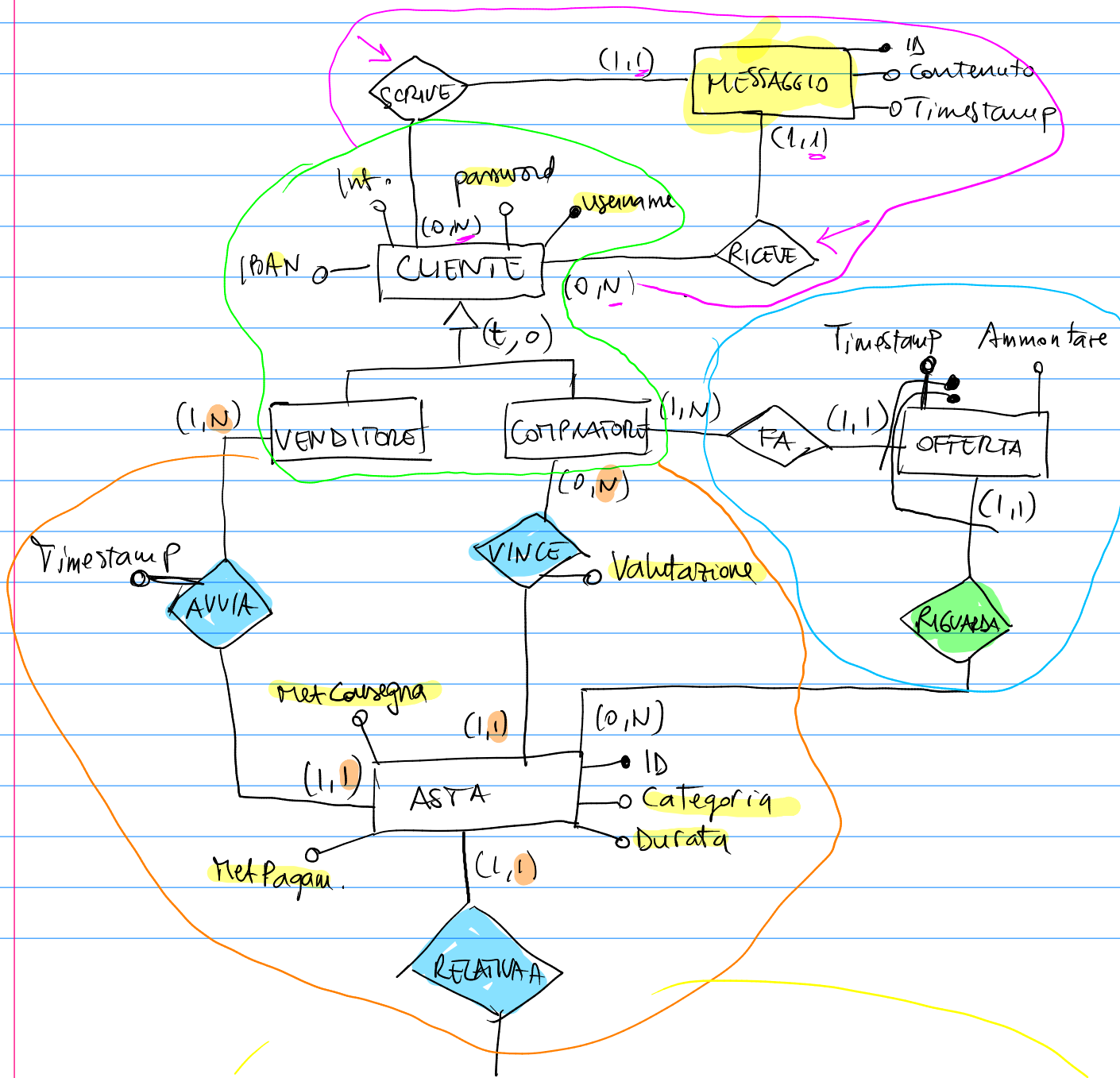
Le gare d'asta sono classificate in categorie, per ciascuna asta corrisponde **una e una sola categoria**.

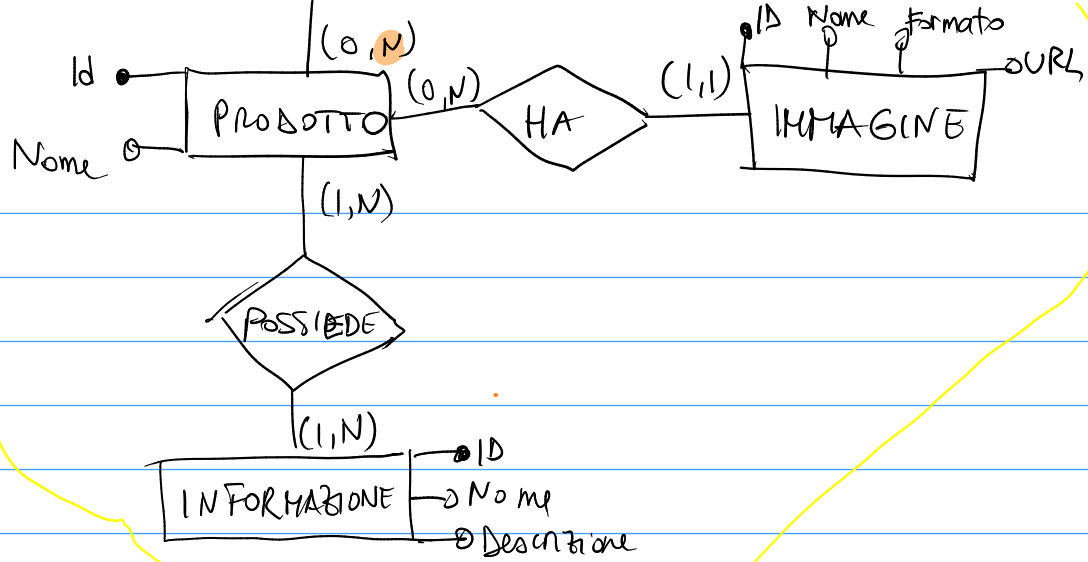
Prima di poter avviare una gara d'asta un cliente deve stabilire una durata in giorni e può caricare diverse immagini (d'esempio, oppure autoscatti). Le stesse immagini possono essere utilizzate in più gare d'asta.

Ciascuna gara d'asta raccoglie informazioni specifiche al prodotto (ad esempio **consumo**, **volume**, etc.), un **metodo di pagamento** e un **metodo di consegna**.

Ogni cliente può **inviare e ricevere messaggi**, tuttavia ciascun messaggio non può essere inviato a più persone.

Ogni cliente è caratterizzato da un profilo di reputazione variabile nel tempo. Al momento della chiusura dell'asta, il vincitore deve valutare il venditore, in modo tale da poter aggiornare il suo profilo.





In una gerarchia parziale NON si può fare un collasso verso il basso !!

CLIENTE (Username, Password, IBAN, Interessatario, Venditore, Comp)

MESSAGGIO (ID, contenuto, Timestamp, Mittente, Destinataria)

OFFERTA (Compratore, Asta, Timestamp, Ammontare)

ASTA (ID, categoria, Durata, Met Gasegn, Met Pagam, Venditore, Vincitore*, Prodotto, Timestamp Creat., Valutat*)

PRODOTTO (ID, Nome)

IMMAGINE (ID, Nome, Formato, URL, Prodotto)

POSSIEDE (Prodotto, Informazione)

INFORMAZIONE (ID, Nome, Descrizione)