

Progetto di Base di Dati *Esercizio*

Specifiche di progetto (1)

- Vogliamo progettare il sistema informativo di supporto ad un sito per l'acquisto di prodotti on-line.
- Il sito potrà essere utilizzato solamente da utenti registrati, quindi anche gli acquisti possono essere effettuati solamente da utenti che si sono preventivamente registrati sul sito. Ovviamente è possibile che un utente sia registrato sul sito senza aver mai effettuato nessun acquisto.
- I fornitori possono vendere i propri prodotti tramite il sito solo nel caso in cui siano preventivamente registrati.
- I prodotti acquistati mediante il sito possono essere consegnati all'acquirente solamente da corrieri che siano preventivamente registrati sul sito.

Specifiche di progetto (2)

- I fornitori offrono i propri prodotti, specificando prezzi e condizioni di vendita e di spedizione.
- Uno stesso prodotto potrà essere venduto da fornitori diversi con prezzi e condizioni di vendita e spedizione diverse.
- Il cliente sceglierà il fornitore da cui comprare un prodotto in base al suo criterio di scelta (a causa del prezzo, delle condizioni di vendita, ecc.).

Specifiche di progetto (3)

- Ciascun corriere metterà a disposizione della clientela vari tipi d consegna.
- Ciascun fornitore registrato sul sito effettuerà accordi commerciali con alcuni dei corrieri registrati sul sito.
- Nel momento in cui un fornitore decida di servirsi di un dato corriere, questi permetterà l'utilizzo di tutti i tipi di consegna messi a disposizione.
- Nel momento in cui un cliente fa un ordine ad un fornitore potrà scegliere per la consegna solamente uno dei corrieri convenzionati con il fornitore. Ovviamente il cliente sceglierà il corriere in base a propri criteri (prezzo, tempi di consegna, garanzie offerte sulla consegna, ecc.).

Specifiche di progetto (4)

- Il sistema informativo dovrà mantenere anche informazioni sull'abbinamento di prodotti appartenenti a categorie diverse.
- Ad esempio, se un cliente acquista un prodotto di categoria X , il sistema informativo dovrà essere in grado di consigliare i prodotti della categoria Y che sono abbinabili con il prodotto appena acquistato.

Specifiche di progetto (5)

- Nel caso in cui il cliente effettui un ordine tramite il sito potrà specificare il proprio grado di soddisfazione sull'operato del fornitore, inserendo nella base di dati un voto compreso tra 0 e 10.
- Analogamente il cliente potrà specificare il proprio grado di soddisfazione sull'operato del corriere.
- Il voto che un cliente dà ad un fornitore o ad un corriere è legato ad un ordine e quindi ad una data.

Specifiche di progetto (6)

- Sulla base di dati progettata dovranno essere effettuate delle query in grado di recuperare almeno le seguenti informazioni:
 - Elenco di tutti i prodotti di una data categoria X che sono consigliati in seguito all'acquisto di un dato prodotto di categoria Y .
 - Elenco dei fornitori che vendono il maggior numero di prodotti diversi di una data categoria X .
 - Elenco dei fornitori che vendono un dato prodotto X a prezzo più basso.
 - Elenco dei fornitori con voto medio più alto tra quelli che hanno un dato prodotto X presente in magazzino.

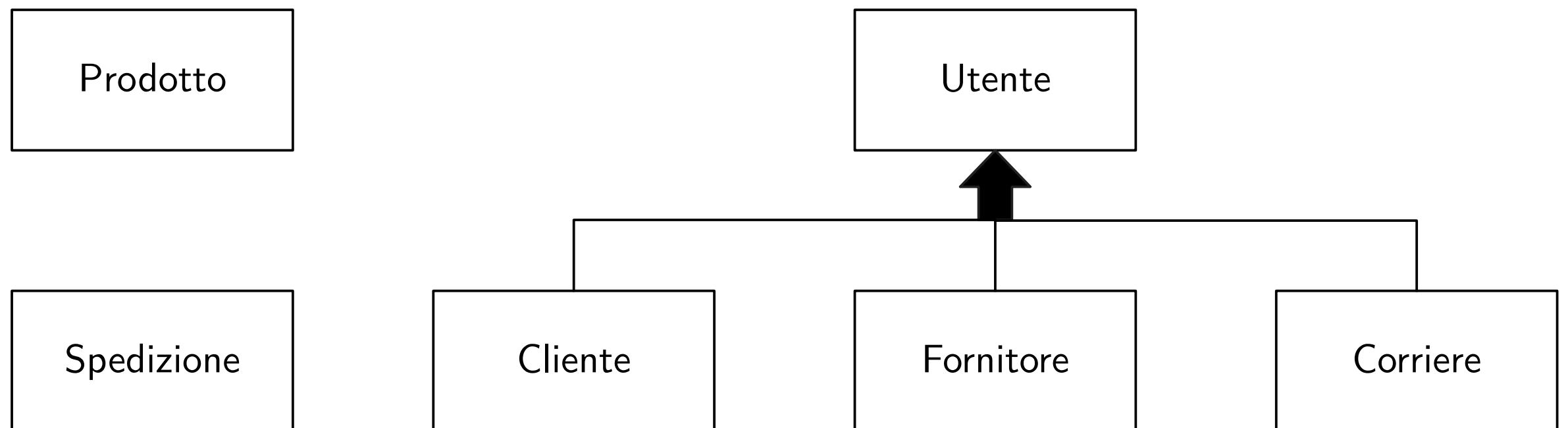
Specifiche di progetto (7)

- Inoltre sulla base di dati dovranno essere effettuate le seguenti operazioni:
 - Inserimento di un nuovo ordine (con grado di soddisfazione del cliente nei confronti dell'operato del fornitore e del corriere nullo).
 - Inserimento del grado di soddisfazione del cliente nei confronti dell'operato del fornitore e/o del corriere relativamente ad un ordine già effettuato.
 - Inserimento di un nuovo prodotto nel magazzino di un fornitore (supponendo che tale prodotto fosse già presente all'interno della base di dati).

Individuazione Entità (1)

- Iniziamo la progettazione individuando le entità della base di dati:
 - Gli **Utenti** sono registrati e costituiscono un'entità
 - Un **Prodotto** è un'entità, la **Categoria** a cui appartiene può essere considerata come suo attributo
 - La **Spedizione** (il tipo)
- Possiamo **specializzare Utente** con **Cliente**, **Fornitore** e **Corriere**
- Otteniamo, quindi, la seguente situazione dove gli **attributi in comune** (Telefono, Indirizzo, e-mail, ecc) di Cliente, Fornitore e Corriere saranno attributi di Utente

Individuazione Entità (2)



Individuazione Entità (3)

- La base di dati deve tenere memoria degli **ordini** effettuati in modo che un voto possa essere assegnato sia ai fornitori che ai corrieri.
- Ciascun ordine consisterà delle seguenti informazioni:
 - Cliente che ha effettuato l'ordine
 - Fornitore che ha ricevuto l'ordine
 - Corriere utilizzato per la consegna dell'ordine
 - Tipo di spedizione utilizzato per recapitare l'ordine
 - Prodotti acquistati tramite l'ordine
 - Voto al corriere
 - Voto al fornitore

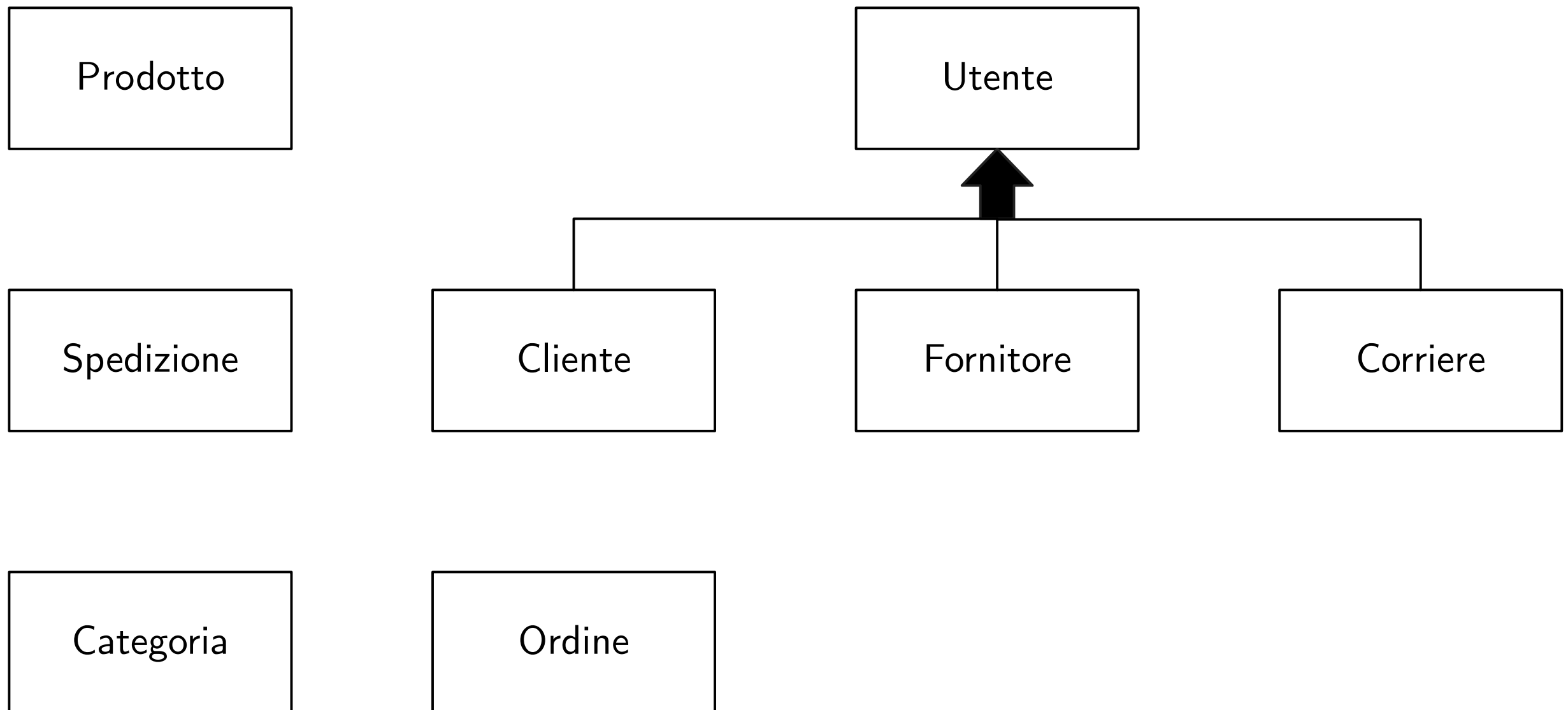
Individuazione Entità (4)

- Poiché un ordine non ha un'esistenza indipendente dalle altre entità (ad esempio un ordine non può esistere se non c'è un cliente che lo genera), possiamo classificare **Ordine** come una ***relationship***
- Ordine stabilisce **un'associazione tra 5 entità diverse**:
 - Cliente
 - Fornitore
 - Corriere
 - Spedizione
 - Prodotti
- Ed ha alcuni **attributi propri**
 - Voto al corriere e voto al fornitore

Individuazione Entità (5)

- Nel ragionamento fin qui fatto abbiamo stabilito due punti che possono risultare difficili da gestire
 1. una ***relationship*** **connessa a 5 entità** diverse pone problemi soprattutto per quanto riguarda l'individuazione delle cardinalità
 - Possiamo pensare di raffinare (approccio top-down) la *relationship* Ordine in un'entità Ordine correlata alle 5 entità Cliente, Fornitore,.. da 5 *relationship* diverse
 2. I prodotti hanno un **attributo Categoria**. Questa soluzione potrebbe dare origine a **inconsistenze in fase di inserimento**. Ad esempio, un fornitore potrebbe inserire
 - Prodotto = "x", Categoria = "RAM"
 - Prodotto = "y", Categoria = "ram"Come conseguenza nella base di dati i due prodotti risulterebbero appartenenti a categorie diverse.
- Per evitare problemi di questo tipo possiamo effettuare ancora un raffinamento e trasformare Categoria in un'entità collegata a Prodotto da una *relationship* di appartenenza.

Individuazione Entità (6)



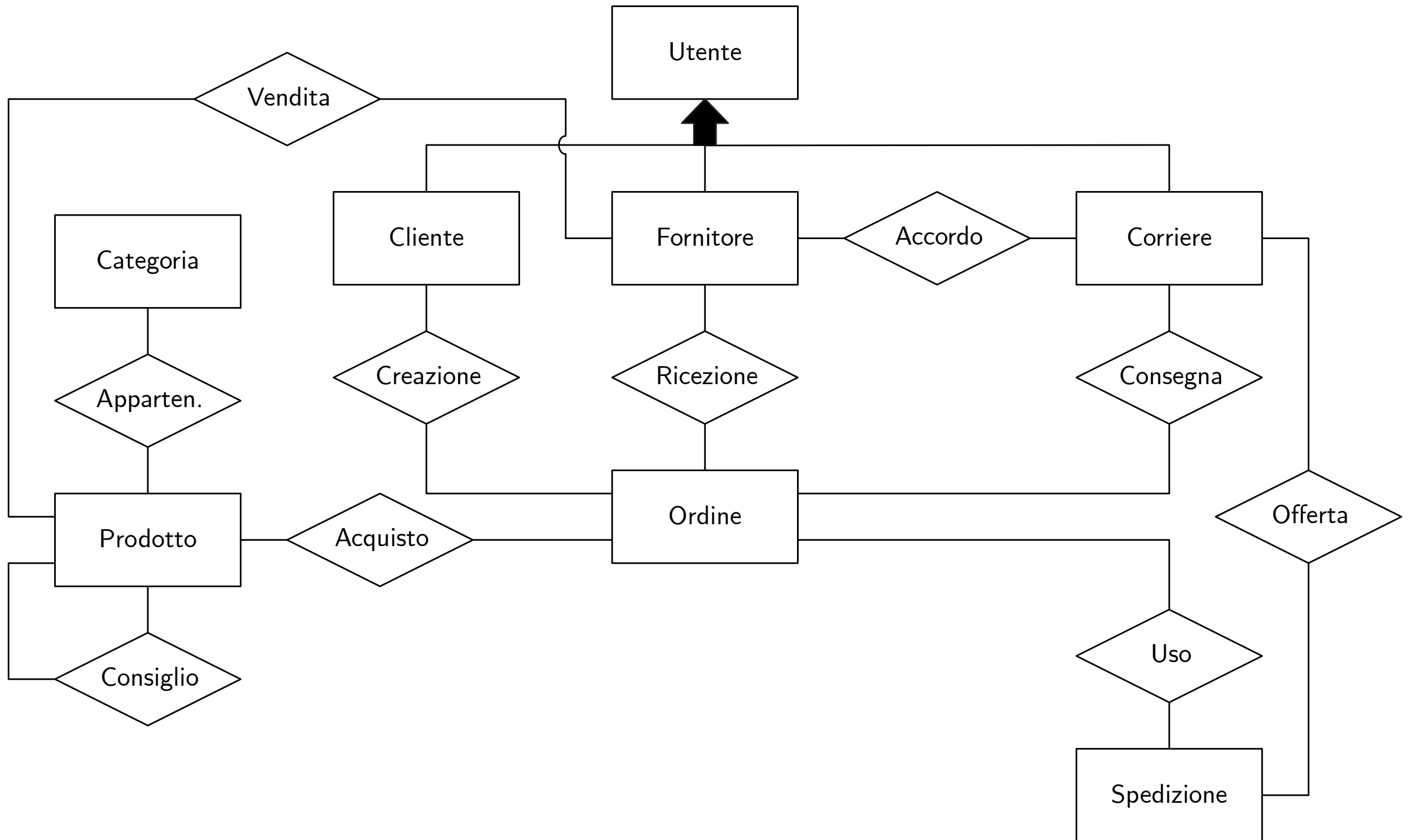
Individuazione *Relationship* (1)

- Vediamo per prime le *relationship* già individuate:
 - Occorre mettere in relazione l'entità Ordine e l'entità Cliente: introduciamo la *relationship* **Creazione**
 - Occorre mettere in relazione l'entità Ordine e l'entità Fornitore: introduciamo la *relationship* **Ricezione**
 - Occorre mettere in relazione l'entità Ordine e l'entità Prodotto: introduciamo la *relationship* **Acquisto**
 - Occorre mettere in relazione l'entità Ordine e l'entità Corriere: introduciamo la *relationship* **Consegna**
 - Occorre mettere in relazione l'entità Ordine e l'entità Spedizione: introduciamo la *relationship* **Uso**
 - Occorre mettere in relazione l'entità Prodotto e l'entità Categoria: introduciamo la *relationship* **Appartenenza**

Individuazione *Relationship* (2)

- Occorre mettere in relazione l'entità Fornitore e l'entità Prodotto: introduciamo la *relationship* **Vendita**
- Occorre mettere in relazione l'entità Fornitore e l'entità Corriere: introduciamo la *relationship* **Accordo**
- Occorre mettere in relazione l'entità Corriere e l'entità Spedizione: introduciamo la *relationship* **Offerta**
- Occorre mettere in relazione l'entità Prodotto con se stessa per poter memorizzare l'abbinamento tra prodotti di categoria diversa: introduciamo la *relationship* **Consiglio**

Individuazione *Relationship* (3)



Individuazione Attributi (1)

- Vediamo quali attributi associare alle varie entità:
 - **Utente:** Indirizzo, Telefono, E-mail, DataRegistrazione, Username, Password
 - **Cliente:** Nome, Cognome, CF
 - **Fornitore:** RagSociale, PartitaIVA
 - **Corriere:** RagSociale, PartitaIVA
 - **Categoria:** CodiceCat, Descrizione
 - **Prodotto:** CodiceProd, Marca, Modello, Descrizione, Foto
 - **Ordine:** CodiceOrd, DataEmissione, DataPagamento, TipoPagamento, DataConsegna, VotoFornitore, VotoCorriere
 - **Spedizione:** AreaCoperta, DimMax, PesoMax

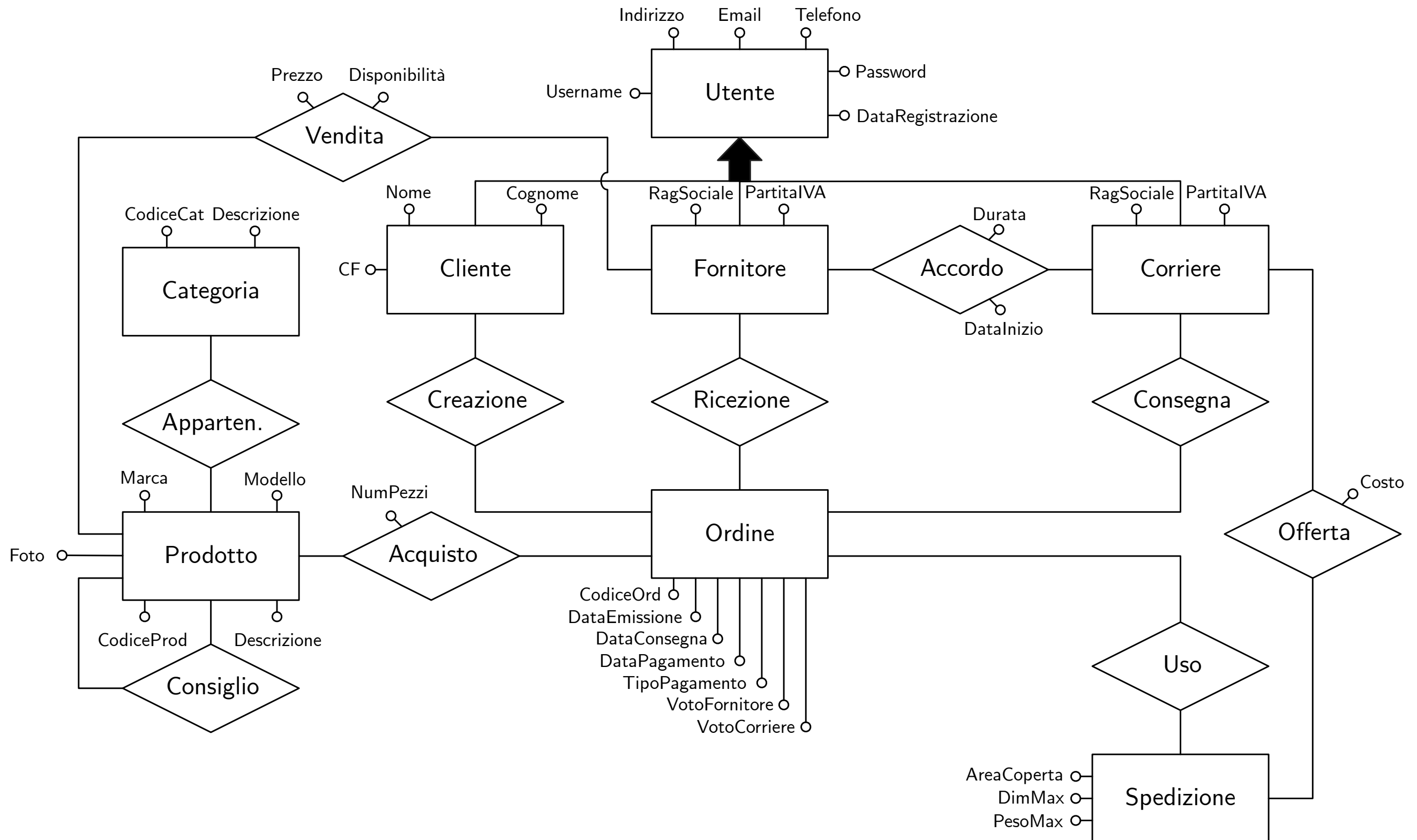
Individuazione Attributi (2)

- Non abbiamo associato il prezzo all'entità Prodotto perché un medesimo prodotto potrebbe essere venduto da fornitori diversi a prezzi diversi.
- Analogamente non abbiamo associato il prezzo all'entità Spedizione perché supponiamo che uno stesso tipo di spedizione possa essere offerto da vari corrieri a prezzi diversi.

Individuazione Attributi (3)

- Vediamo quali attributi potremmo associare alle varie *relationship*:
 - **Vendita**: Prezzo, Disponibilità
 - **Appartenenza**: nessun attributo
 - **Creazione**: nessun attributo
 - **Ricezione**: nessun attributo
 - **Acquisto**: NumPezzi
 - **Consiglio**: nessun attributo
 - **Accordo**: DataInizio, Durata
 - **Consegna**: nessun attributo
 - **Uso**: nessun attributo
 - **Offerta**: Costo
- Alla *relationship* Accordo abbiamo associato l'attributo DataInizio e Durata, ma supponiamo di non dover fare nessun controllo perché manterremo nella base di dati solamente gli accordi attualmente in vigore.
- Nella *relationship* Acquistati manteniamo memoria del numero di pezzi di ciascun prodotto richiesti in ciascun ordine.

Individuazione Attributi (4)



Individuazione Cardinalità (1)

- Vediamo adesso le varie cardinalità:
 - Entità **Ordine** - *Relationship* **Ricezione**: (1,1) in quanto un ordine viene ricevuto da un solo fornitore.
 - Entità **Ordine** - *Relationship* **Consegna**: (1,1) in quanto un ordine viene recapitato da un solo corriere.
 - Entità **Ordine** - *Relationship* **Creazione**: (1,1) in quanto un ordine viene eseguito da un solo cliente.
 - Entità **Ordine** - *Relationship* **Uso**: (1,1) in quanto per la consegna di un ordine si utilizza un solo tipo di spedizione.
 - Entità **Ordine** - *Relationship* **Acquisto**: (1,N) in quanto un ordine può contenere più prodotti diversi.

Individuazione Cardinalità (2)

- Entità **Fornitore** – *Relationship Ricezione*: (0,N) in quanto un fornitore può aver ricevuto più ordini o nessun ordine (se ad esempio il fornitore si è appena iscritto al sito).
- Entità **Fornitore** – *Relationship Accordo*: (1,N) si suppone che, contemporaneamente all'iscrizione al sito, ogni fornitore si accordi con almeno un corriere.
- Entità **Fornitore** – *Relationship Vende*: (1,N) un fornitore può vendere più di un prodotto. Si suppone che, nel momento in cui si iscrive al sito, ogni fornitore sia in grado di vendere almeno un prodotto.

Individuazione Cardinalità (3)

- Entità **Corriere** – *Relationship* **Accordo**: (0,N) un corriere può essere accordato con più fornitori o con nessun fornitore (ad esempio se si è appena iscritto al sito).
- Entità **Corriere** – *Relationship* **Consegna**: (0,N) un corriere può aver consegnato più ordini o nessun ordine (ad esempio se si è appena iscritto al sito).

Individuazione Cardinalità (4)

- Entità **Spedizione** – *Relationship Uso*: (0,N) un tipo di spedizione può essere utilizzato per consegnare più ordini. È possibile però che un tipo di spedizione non sia mai stato utilizzato (ad esempio se è stato appena introdotto nel sito o se è particolarmente svantaggioso).
- Entità **Spedizione** – *Relationship Offerta*: (1,N) un tipo di spedizione può essere offerto da più corrieri. Ciascun tipo di spedizione, però, deve essere offerto da almeno un corriere, altrimenti non avrebbe motivo di essere inserito nella base di dati del sito.

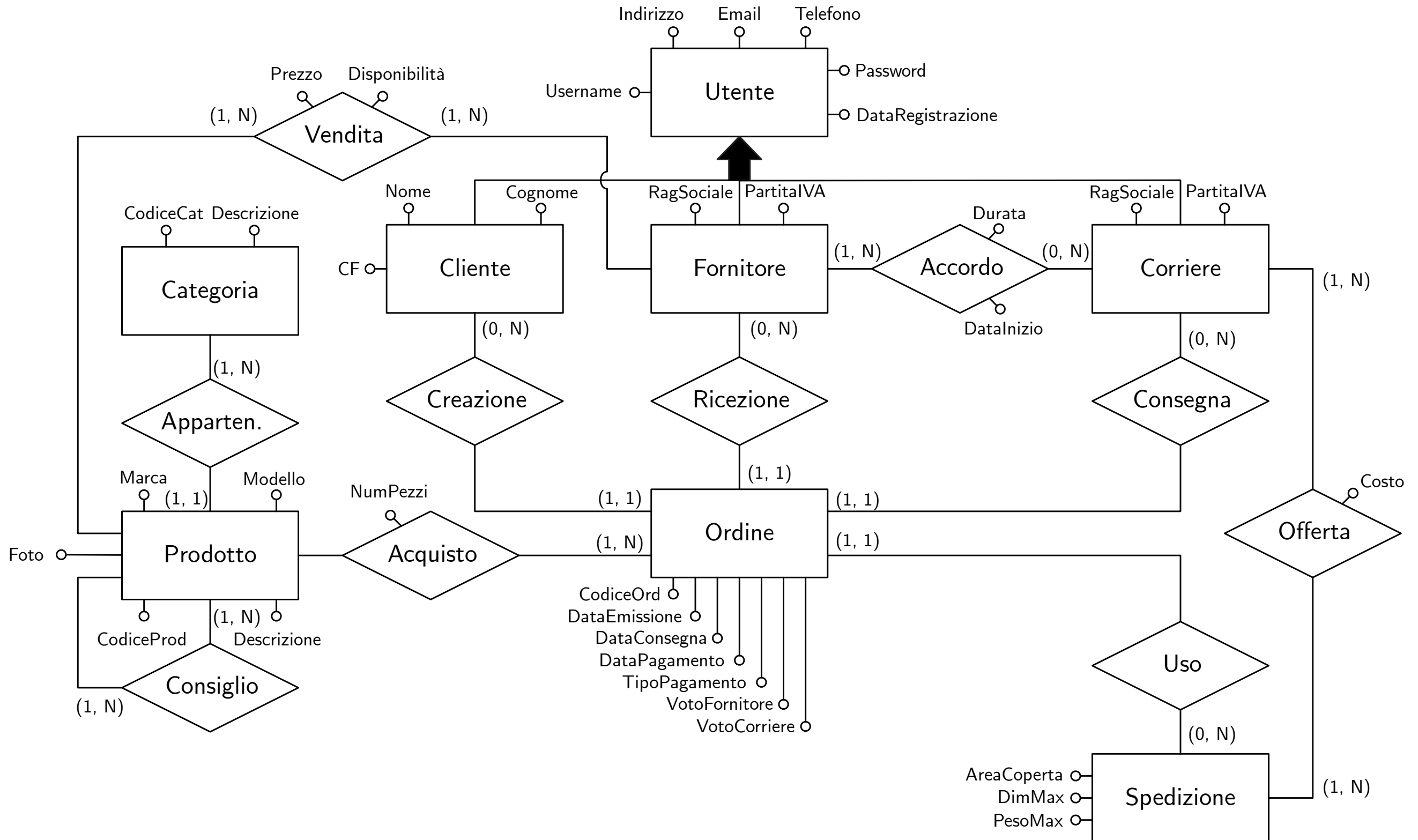
Individuazione Cardinalità (5)

- Entità **Cliente** – *Relationship Ordine*: (0,N) ciascun cliente può aver effettuato più ordini. È possibile che un cliente non abbia effettuato alcun ordine.
- Entità **Prodotto** – *Relationship Vendita*: (1,N) un prodotto può essere venduto da più fornitori, e deve essere venduto da almeno un fornitore (altrimenti non avrebbe senso la presenza del prodotto nella base di dati del sito).
- Entità **Corriere** – *Relationship Offerta*: (1,N) un corriere può offrire tipi di spedizione diversi. Si suppone che un corriere offra almeno un tipo di spedizione.

Individuazione Cardinalità (6)

- Entità **Prodotto** – *Relationship* **Consiglio**: (1,N)
ciascun prodotto ha almeno un altro prodotto consigliato.
- Entità **Categoria** – *Relationship* **Appartenenza**:
(1,N) la base di dati deve contenere almeno un prodotto per ciascuna categoria, infatti l'esistenza di una categoria vuota non avrebbe senso.
- Entità **Prodotto** – *Relationship* **Appartenenza**:
(1,1) ciascun prodotto appartiene ad una ed una sola categoria.

Schema ER



Individuazione Nuovi Attributi (1)

- Se analizziamo le operazioni che devono essere compiute sulla base di dati ci rendiamo conto che potrebbe convenire introdurre dei nuovi attributi al fine di facilitarle
- Questi attributi possono anche rappresentare informazione già presente nella base di dati.
- Consideriamo, ad esempio:
 - Elenco dei fornitori con voto medio più alto tra quelli che hanno un dato prodotto X presente in magazzino.
- In questo caso per recuperare il voto medio di un fornitore dobbiamo andare a leggere tutti gli ordini ricevuti da tale fornitore.

Individuazione Nuovi Attributi (2)

- Per facilitare questa operazione potremmo introdurre i seguenti attributi sull'entità Fornitore:
 - TotaleVoti
 - NumeroVoti
- In questo modo, ogni volta che introduciamo un ordine dovremo anche aggiornare questi attributi per il fornitore che ha ricevuto l'ordine.

Individuazione Nuovi Attributi (3)

- Quindi le operazioni coinvolte dall'introduzione di questi nuovi attributi sono le seguenti:
 - Elenco dei fornitori con voto medio più alto tra quelli che hanno un dato prodotto X presente in magazzino.
 - Inserimento del grado di soddisfazione del cliente nei confronti dell'operato del fornitore e/o del corriere in un ordine di cui si conosce il codice.
- Per capire se l'introduzione dei nuovi attributi è vantaggiosa in termini di tempi di esecuzione, dobbiamo valutare i diversi costi per queste due operazioni.

Tavola dei volumi (1)

- Per prima cosa è necessario valutare il numero di occorrenze di una data entità o *relationship*.
- Per fare questo dobbiamo basarci sulle indicazioni presenti all'interno delle specifiche.
- Nel caso in cui queste non siano sufficienti, dovremo fare delle ulteriori ipotesi che faranno parte della descrizione finale della base di dati.

Specifiche di progetto (8)

- Al fine di prendere decisioni sulla struttura della base di dati, si supponga che la base di dati contenga:
 - 600 prodotti diversi
 - 50 fornitori registrati
 - 1500 clienti registrati
 - 15 corrieri registrati
 - 3 abbinamenti (in media) tra un prodotto (di categoria X) e i prodotti della categoria Y .
 - 5 ordini finora effettuati per ciascun prodotto.

Tavola dei volumi (2)

- **Cliente:** 1500 occorrenze (da specifiche)
- **Fornitore:** 50 occorrenze (da specifiche)
- **Corriere:** 15 occorrenze (da specifiche)
- **Utente:** 1565 occorrenze
 - Perché le persone che sono iscritte sia come clienti che come fornitori e/o corrieri hanno più di un account. Quindi gli utenti possono essere al più: $1565 = 1500 + 50 + 15$
- **Prodotto:** 600 occorrenze (da specifiche)
- **Categoria:** 30 occorrenze (supponiamo che la base di dati contenga 30 categorie diverse di prodotti)
- **Spedizione:** 6 occorrenze (supponiamo che la base di dati contenga 6 tipi di spedizione diversi)

Tavola dei volumi (3)

- **Ordine:** 3000 occorrenze (perché da specifiche vengono effettuati in media 5 ordini per ciascun prodotto: $3000 = 600 \times 5$)
- **Vendita:** 1800 occorrenze (supponiamo che ciascun fornitore venda in media 36 prodotti diversi: $1800 = 50 \times 36$)
- **Accordo:** 150 occorrenze (supponiamo che ciascun fornitore si sia accordato in media con 3 corrieri diversi: $150 = 50 \times 3$)
- **Appartenenza:** 600 occorrenze (perché ciascun prodotto appartiene ad una, ed una sola, categoria)
- **Creazione:** 3000 occorrenze (perché ciascun ordine è eseguito da un solo cliente)
- **Ricezione:** 3000 occorrenze (perché ciascun ordine è ricevuto da un solo fornitore)

Tavola dei volumi (4)

- **Consegna:** 3000 occorrenze (perché ciascun ordine è recapitato da un solo corriere)
- **Uso:** 3000 occorrenze (perché ciascun ordine è recapitato utilizzando un solo tipo di consegna)
- **Offerta:** 60 occorrenze (perché ciascun corriere offre in media 4 tipi di spedizione diversi: $60 = 15 \times 4$)
- **Acquisto:** 6000 occorrenze (perché ciascun ordine comprende in media 2 prodotti diversi: $6000 = 3000 \times 2$)
- **Consiglio:** 1800 occorrenze (perché da specifiche, ciascun prodotto ha in media 3 abbinamenti)

Costo delle operazioni (1)

- Date le operazioni coinvolte:
 - **Operazione 1:** Elenco dei fornitori con voto medio più alto tra quelli che hanno un dato prodotto X presente in magazzino.
 - **Operazione 2:** Inserimento del grado di soddisfazione del cliente nei confronti dell'operato del fornitore e/o del corriere per un ordine di cui si conosce il codice.
- Dobbiamo individuare le **operazioni elementari** nei seguenti casi:
 - svolgimento dell'operazione 1 **senza** nuovi attributi
 - svolgimento dell'operazione 1 **con** nuovo attributo
 - svolgimento dell'operazione 2 **senza** nuovi attributi
 - svolgimento dell'operazione 2 **con** nuovi attributi

Costo delle operazioni (2)

- Esecuzione Operazione 1 senza attributi aggiuntivi:
 - 1 lettura in Prodotto per recuperare il codice del prodotto
 - 3 letture in Vendita per recuperare il codice dei fornitori che vendono quel prodotto (un prodotto è venduto in media da 3 fornitori: $3 = 1800 / 600$).
 - 180 letture in Ricezione per recuperare il codice degli ordini ricevuti dai fornitori che hanno quel prodotto in magazzino (ciascun fornitore riceve in media 60 ordini: $60 = 3000 / 50$).
 - 180 letture in Ordine per poter leggere il voto assegnato nei vari ordini ai fornitori che hanno quel prodotto in magazzino.

Costo delle operazioni (3)

- Esecuzione Operazione 1 con nuovi attributi:
 - 1 lettura in Prodotto per recuperare il codice del prodotto
 - 3 letture in Vendita per recuperare il codice dei fornitori che vendono quel prodotto (un prodotto è venduto in media da 3 fornitori: $3 = 1800 / 600$).
 - 3 letture in Fornitore per recuperare il TotaleVoti e il NumeroVoti di ciascun fornitore che ha quel prodotto in magazzino.

Costo delle operazioni (4)

- Esecuzione Operazione 2 senza attributi aggiuntivi:
 - 1 scrittura in Ordine per inserire un voto non nullo
- Esecuzione Operazione 2 con attributi aggiuntivi:
 - 1 scrittura in Ordine
 - 1 lettura in Ricezione per recuperare il fornitore che ha ricevuto l'ordine
 - 1 lettura in Fornitore per recuperare il TotaleVoti e NumeroVoti
 - 1 scrittura in Fornitore per scrivere il nuovo valore di TotaleVoti e NumeroVoti

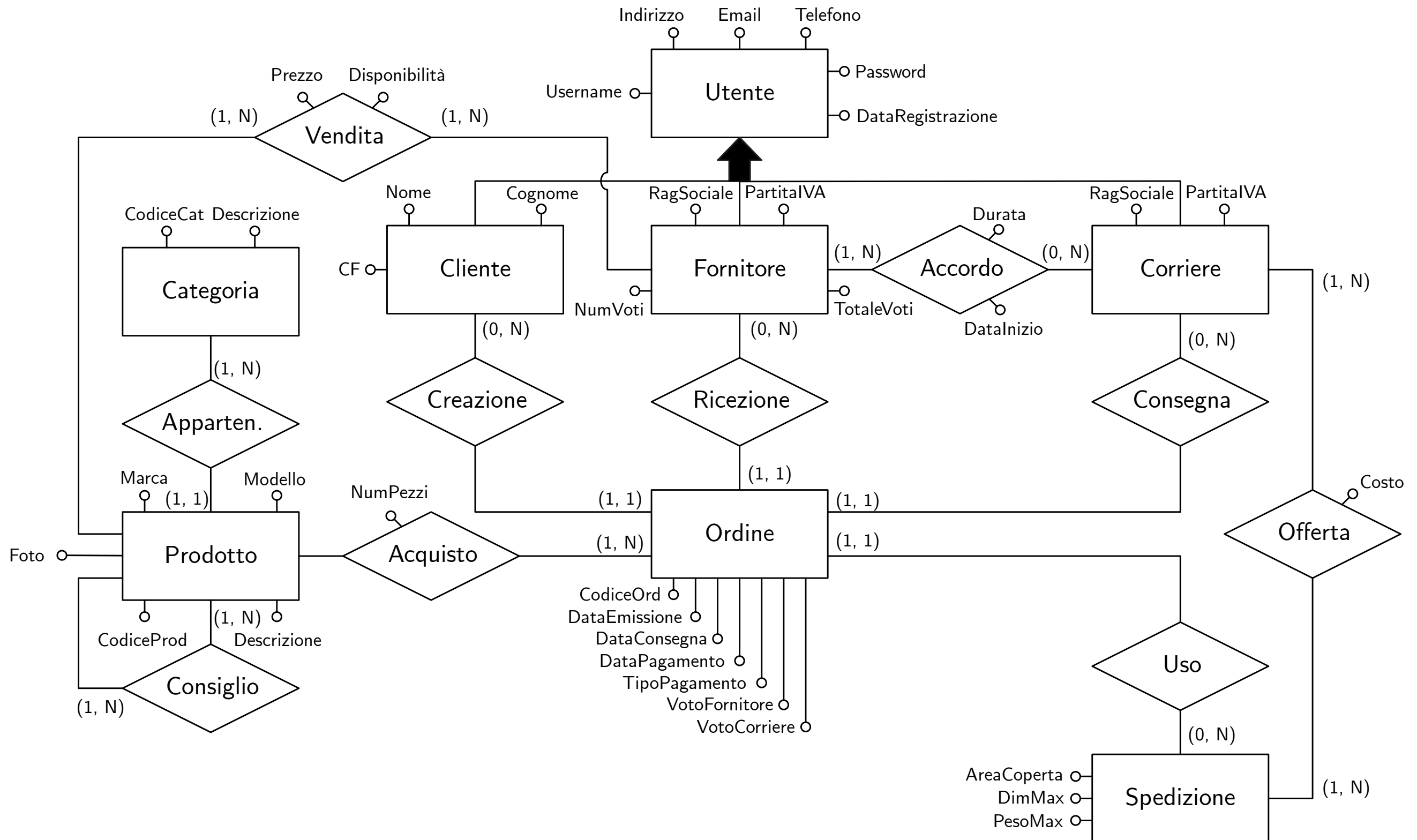
Costo delle operazioni (5)

- Operazioni necessarie per svolgere l'operazione 1 :
 - 363 letture \rightarrow 363 operazioni elementari
- Operazioni necessarie per svolgere l'operazione 1 con attributi aggiuntivi
 - 6 letture \rightarrow 6 operazioni elementari
- Operazioni necessarie per svolgere l'operazione 2:
 - 1 scrittura \rightarrow 2 operazioni elementari
- Operazioni necessarie per svolgere l'operazione 2 con attributi aggiuntivi
 - 2 scritture \rightarrow 4 operazioni elementari
 - 2 letture \rightarrow 2 operazioni elementari

Costo delle operazioni (6)

- Supponendo che l'operazione 2 venga compiuta 10 volte più frequentemente rispetto all'operazione 1, avremo
 - con i nuovi attributi:
 - $6 \text{ Op. Elementari} + 6 \times 10 \text{ Op. Elementari}$
 - $\text{totale} = 66 \text{ Op. Elementari}$
 - senza nuovi attributi:
 - $363 \text{ Op. Elementari} + 2 \times 10 \text{ Op. Elementari}$
 - $\text{totale} = 383 \text{ Op. Elementari.}$
- Al fine di ridurre le operazioni elementari introduciamo i nuovi attributi all'interno dello schema ER.

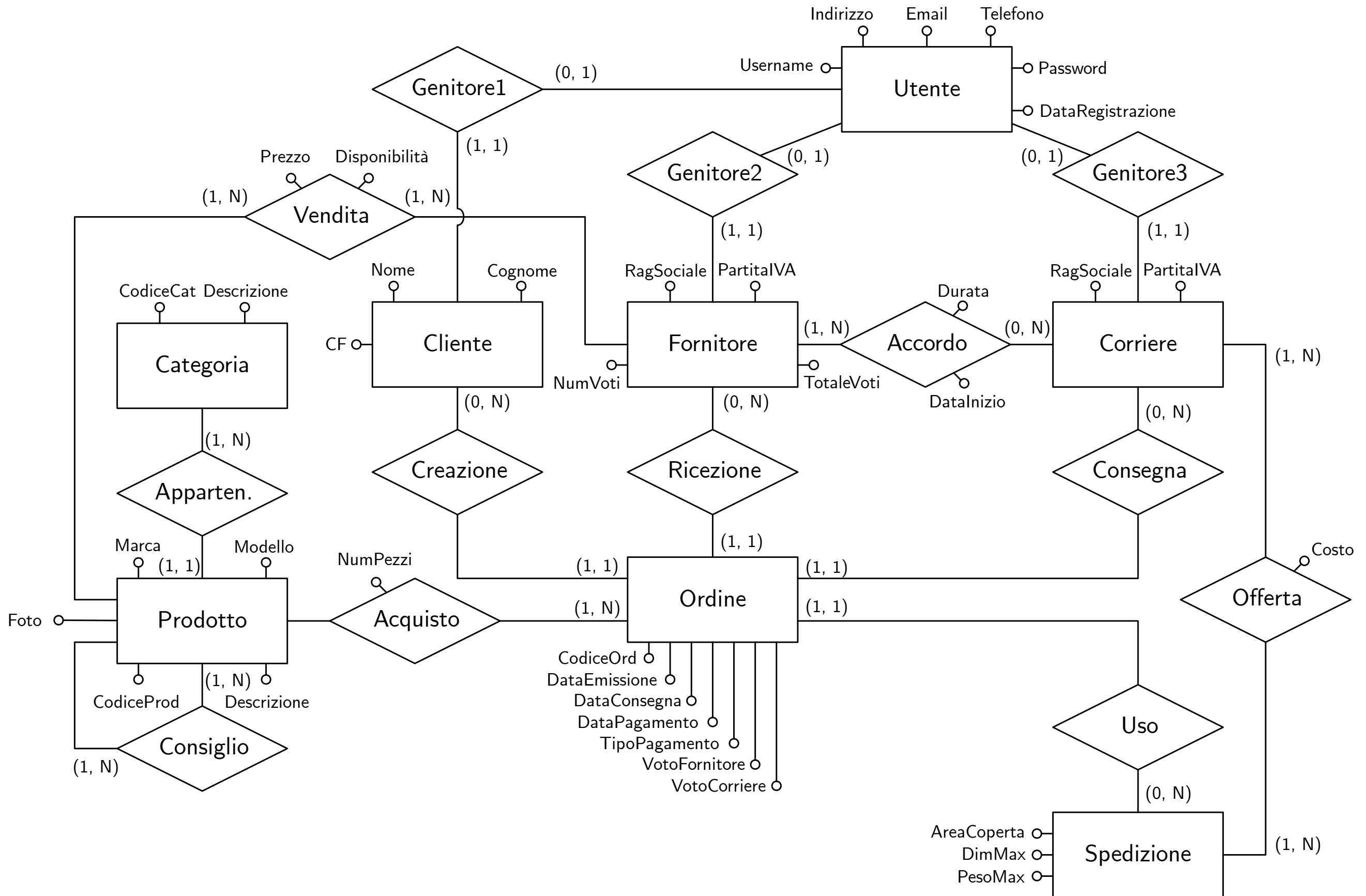
Nuovo Schema ER



Eliminazione delle generalizzazioni (1)

- Vediamo adesso come possiamo **eliminare la generalizzazione** presente nello schema ER
- Poiché si tratta di una **generalizzazione totale**, abbiamo **tre possibilità**:
 - Lasciare padri e figli separati e collegarli **utilizzando alcune *relationship***
 - **Accorpare l'entità padre sulle entità figlie**
 - **Accorpare le entità figlie sull'entità padre**
- Analizziamo gli schemi ER che otteniamo in ciascuno di questi tre casi in modo da **decidere quale** di questi **utilizzare**

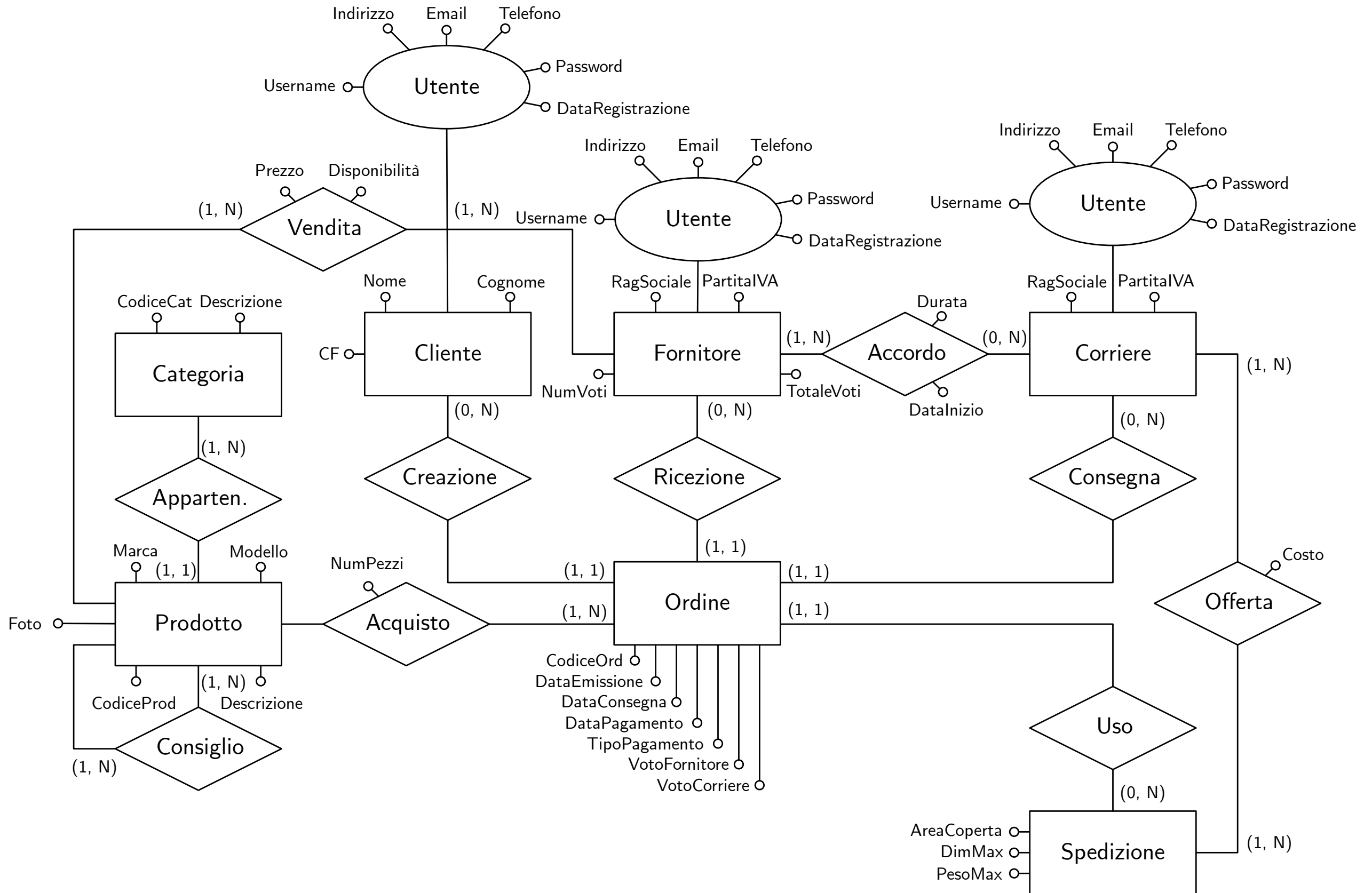
Uso di *relationship*



Eliminazione delle generalizzazioni (2)

- In questo modo abbiamo introdotto nello schema ER **tre nuove *relationship***
- Non abbiamo introdotto **nessun valore NULL** all'interno della base di dati
- Abbiamo mantenuto un **numero non troppo elevato di attributi** per ciascuna entità
 - Quanto più il **numero** di **attributi** di un'entità è **elevato**, tanto più la **gestione** della relativa **tabella** diventa **pesante**
 - È quindi **buona norma** tentare di **limitare il numero di attributi** di ciascuna entità

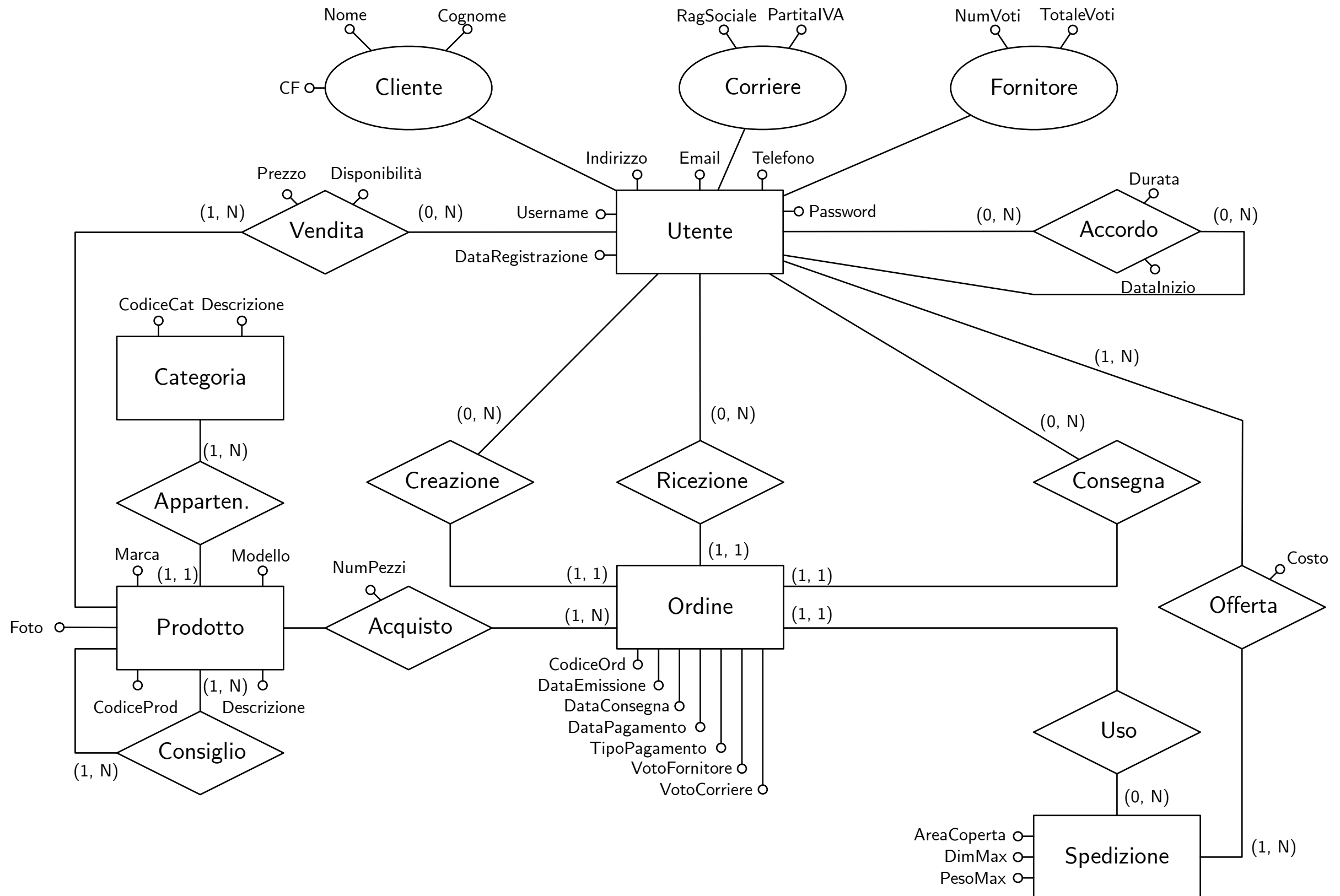
Accorpare padre sulle figlie



Eliminazione delle generalizzazioni (3)

- In questo modo **non abbiamo introdotto** nello schema nessuna **nuova *relationship***
- **Non abbiamo introdotto** nessun **valore NULL** all'interno della base di dati
- Abbiamo però **aggiunto** alle entità Cliente, Fornitore e Corriere **molti nuovi attributi** (evidenziati in rosso nello schema ER) ottenendo **tabelle più pesanti** da gestire
- Inoltre, quando vogliamo controllare la correttezza di una coppia (username, password) dovremo controllare le coppie (username, password) presenti in tre diverse tabelle (a meno che non si sappia in anticipo se tale coppia appartiene ad un cliente, ad un fornitore o ad un corriere)

Accorpare figlie sul padre



Eliminazione delle generalizzazioni (4)

- In questo modo non abbiamo introdotto nello schema ER **nessuna nuova *relationship***
- Abbiamo però introdotto **alcuni valori NULL** all'interno dell'entità Utenti
- Il **numero di attributi** dell'entità Utenti è **cresciuto molto**. La **tabella Utenti è molto pesante da gestire**.
- Notare che anche se un fornitore vende necessariamente almeno un prodotto (cardinalità (1,N) tra la relazione Vende e l'entità Fornitori), non è detto che un utente (che può essere anche un cliente) venda necessariamente un prodotto (quindi cardinalità (0,N) tra la relazione Vende e l'entità Utenti)
- Lo stesso vale per la relazione Accordo

Eliminazione delle generalizzazioni (5)

- Viste le tre possibili soluzioni, possiamo concludere che in questo caso conviene **tradurre la generalizzazione** lasciando l'entità padre e le entità figlie separate e **collegate tramite *relationship***
- In questo modo **non introduciamo** nessun **valore NULL** e il **numero di attributi** di ciascuna entità resta **sufficientemente limitato**

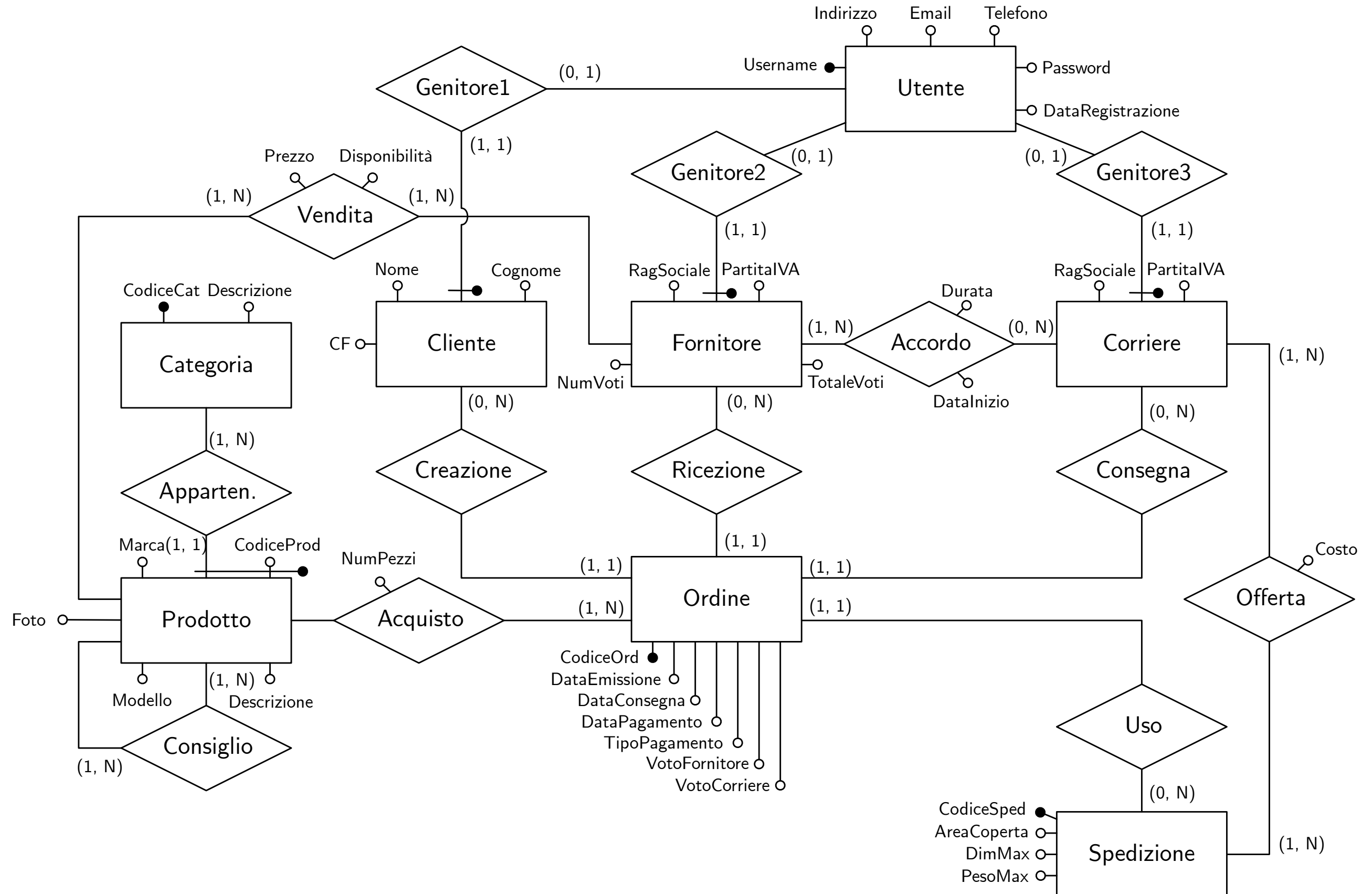
Specifica Identificatori (1)

- Adesso dobbiamo specificare gli identificatori delle varie entità:
 - **Categoria:** utilizziamo CodiceCat come identificatore
 - **Prodotto:** utilizziamo come chiave la coppia CodiceProd (attributo di prodotto) e CodiceCat (identificatore dell'entità Categoria)
 - **Ordine:** utilizziamo CodiceOrd come identificatore
 - **Utente:** utilizziamo Username come identificatore, in quanto ogni utente è caratterizzato dal proprio username
 - **Fornitore:** potremmo utilizzare PartitaIVA come identificatore, in quanto ogni fornitore è caratterizzato dalla propria partita IVA.
 - In questo caso, però, utilizzeremo come identificatore lo Username dell'utente associato

Specifica Identificatori (2)

- **Corriere:** potremmo utilizzare PartitaIVA come identificatore, in quanto ogni corriere è caratterizzato dalla propria partita IVA
 - In questo caso, però, utilizzeremo come identificatore lo Username dell'utente associato
- **Cliente:** potremmo utilizzare CF come identificatore, in quanto ogni corriere è caratterizzato dal proprio codice fiscale
 - In questo caso, però, utilizzeremo come identificatore lo Username dell'utente associato
- **Spedizione:** per identificare la spedizione dovremmo utilizzare gli attributi AreaCoperta, DimMax e PesoMax
 - Poiché utilizzare tre attributi come identificatore può risultare scomodo, possiamo aggiungere un attributo CodiceSped ed utilizzarlo come identificatore

Specifica Identificatori (3)



Traduzione in tabelle (1)

- Traduciamo lo schema ER ottenuto in tabelle
- Ciascuna entità viene tradotta in una tabella a sé stante
- Vediamo come tradurre le varie *relationship*:
 - **Genitore1**: la traduciamo accorpando la tabella sull'entità Cliente (cardinalità (1,1) tra la *relationship* Genitore1 e l'entità Cliente)
 - **Genitore2**: la traduciamo accorpando la tabella sull'entità Fornitore (cardinalità (1,1) tra la *relationship* Genitore2 e l'entità Fornitore)
 - **Genitore3**: la traduciamo accorpando la tabella sull'entità Corriere (cardinalità (1,1) tra la *relationship* Genitore3 e l'entità Corriere)
 - **Vendita**: la traduciamo in una tabella a sé stante (*relationship* da N a N)

Traduzione in tabelle (2)

- **Accordo:** la traduciamo in una tabella a sé stante (cardinalità (1,N) tra la *relationship* Accordo e l'entità Fornitore e cardinalità (0,N) tra la *relationship* Accordo e l'entità Corriere)
- **Appartenenza:** la traduciamo accorpando la tabella sull'entità Prodotto (cardinalità (1,1) tra la *relationship* Appartiene e l'entità Prodotto)
- **Creazione:** la traduciamo accorpando la tabella sull'entità Ordine (cardinalità (1,1) tra la *relationship* Crea e l'entità Ordine)
- **Ricezione:** la traduciamo accorpando la tabella sull'entità Ordine (cardinalità (1,1) tra la *relationship* Riceve e l'entità Ordine)
- **Consegna:** la traduciamo accorpando la tabella sull'entità Ordine (cardinalità (1,1) tra la *relationship* Consegna e l'entità Ordine)

Traduzione in tabelle (3)

- **Uso:** la traduciamo accorpando la tabella sull'entità Ordine (cardinalità (1,1) tra la *relationship* Uso e l'entità Ordine)
- **Acquisto:** la traduciamo in una tabella a sé stante (cardinalità (1,N) tra la *relationship* Acquisto e l'entità Ordine e cardinalità (0,N) tra la *relationship* Acquisto e l'entità Prodotto)
- **Offerta:** la traduciamo in una tabella a sé stante (*relationship* da N a N)
- **Consiglio:** la traduciamo in una tabella a sé stante (*relationship* da N a N)

Tabelle (1)

- **Utente**(Username, DataRegistrazione, Telefono, E-mail, Indirizzo, Password)
- **Cliente**(Username, Nome, Cognome, CF)
- **Fornitore**(Username, RagSociale, PartitaIVA, NumVoti, TotaleVoti)
- **Corriere**(Username, RagSociale, PartitaIVA)
- **Spedizione**(CodiceSped, AreaCoperta, DimMax, PesoMax)
- **Ordine**(CodiceOrd, DataEmissione, TipoPagamento, DataPagamento, DataConsegna, VotoFornitore, VotoCorriere, CodiceCli, CodiceFo, CodiceCor, CodiceSped)

Tabelle (2)

- **Categoria**(CodiceCat, Descrizione)
- **Prodotto**(CodiceProd, CodiceCat, Foto, Marca, Descrizione, Modello)
- **Vendita**(CodiceFor, CodiceProd, CodiceCat, Prezzo, DispMagazzino)
- **Consiglio**(CodiceCat1, CodiceProd1, CodiceCat2, CodiceProd2)
- **Acquisto**(CodiceProd, CodiceCat, CodiceOrd, NumPezzi)
- **Offerta**(CodiceCorr, CodiceSped, Costo)

Vincoli di integrità referenziale (1)

- Nel tradurre lo schema ER in tabelle, sono stati introdotti i seguenti vincoli di integrità referenziale:
 - tra l'attributo **Username** della tabella **Cliente** e l'attributo **Username** della tabella **Utente**
 - tra l'attributo **Username** della tabella **Fornitore** e l'attributo **Username** della tabella **Utente**
 - tra l'attributo **Username** della tabella **Corriere** e l'attributo **Username** della tabella **Utente**
 - tra l'attributo **CodiceCli** della tabella **Ordine** e l'attributo **Username** della tabella **Cliente**
 - tra l'attributo **CodiceFor** della tabella **Ordine** e l'attributo **Username** della tabella **Fornitore**
 - tra l'attributo **CodiceCor** della tabella **Ordine** e l'attributo **Username** della tabella **Corriere**

Vincoli di integrità referenziale (2)

- tra l'attributo **CodiceSped** della tabella **Ordine** e l'attributo **CodiceSped** della tabella **Spedizione**
- tra l'attributo **CodiceCat** della tabella **Prodotto** e l'attributo **CodiceCat** della tabella **Categoria**
- tra l'attributo **CodiceCat** della tabella **Prodotto** e l'attributo **CodiceCat** della tabella **Categoria**
- tra l'attributo **CodiceFor** della tabella **Vendita** e l'attributo **Username** della tabella **Fornitore**
- tra l'attributo **CodiceProd** della tabella **Vendita** e l'attributo **CodiceProd** della tabella **Prodotto**
- tra l'attributo **CodiceCat** della tabella **Vendita** e l'attributo **CodiceCat** della tabella **Categoria**

Vincoli di integrità referenziale (3)

- tra l'attributo **CodiceCat** della tabella **Consiglio** e l'attributo **CodiceCat** della tabella **Categoria**
- tra l'attributo **CodiceProd1** della tabella **Consiglio** e l'attributo **CodiceProd** della tabella **Prodotto**
- tra l'attributo **CodiceCat2** della tabella **Consiglio** e l'attributo **CodiceCat** della tabella **Categoria**
- tra l'attributo **CodiceProd2** della tabella **Consiglio** e l'attributo **CodiceProd** della tabella **Prodotto**
- tra l'attributo **CodiceProd** della tabella **Acquisto** e l'attributo **CodiceProd** della tabella **Prodotto**

Vincoli di integrità referenziale (4)

- tra l'attributo **CodiceCat** della tabella **Acquisto** e l'attributo **CodiceCat** della tabella **Categoria**
- tra l'attributo **CodiceOrd** della tabella **Acquisto** e l'attributo **CodiceOrd** della tabella **Ordine**
- tra l'attributo **CodiceCor** della tabella **Offerta** e l'attributo **Username** della tabella **Corriere**
- tra l'attributo **CodiceSped** della tabella **Offerta** e l'attributo **CodiceSped** della tabella **Spedizione**