



Esercizi svolti a lezione

Sistemi Informativi L-B

Home Page del corso:

<http://www-db.deis.unibo.it/courses/SIL-B/>

Versione elettronica: [esercizi.pdf](#)

Versione aggiornata al 25/05/2010



Modello E/R (1.1)

Dal compito del 28/03/2007:

- *Nonsapevo.it è un nuovo sito che permette di segnalare notizie interessanti e di condividerle con tutti. Le informazioni necessarie al servizio vengono registrate in un database.*
 - Ogni utente può inserire delle notizie, indicando per ognuna un titolo e un link (esterno) di collegamento alla notizia completa. Eventualmente si può indicare anche una piccola frase che introduce la notizia. Gli utenti possono collaborare solo se sono registrati al sito (gratuitamente). Per ogni account si registra Username, Password, Nome, Cognome e email.
 - Chi accede al sito legge le notizie e (registrandosi) può indicare un parere positivo o negativo (non più di uno per ogni notizia). Gli utenti registrati possono anche lasciare commenti alle notizie che leggono (anche più commenti per la stessa notizia), e indicare le proprie notizie preferite. Il sistema registra data e ora di inserimento per ogni notizia, per ogni voto e per ogni commento.
 - Per facilitarne la gestione, ad ogni notizia vengono associate alcune parole chiave, o “tag” (almeno una) scelte dall’autore. Chi si dichiara interessato ad alcuni tag può ricevere periodicamente via email le notizie relative.
-
- Quali sono le principali entità derivabili dalle specifiche?
 - Quali le associazioni?



Modello E/R (1.2): entità

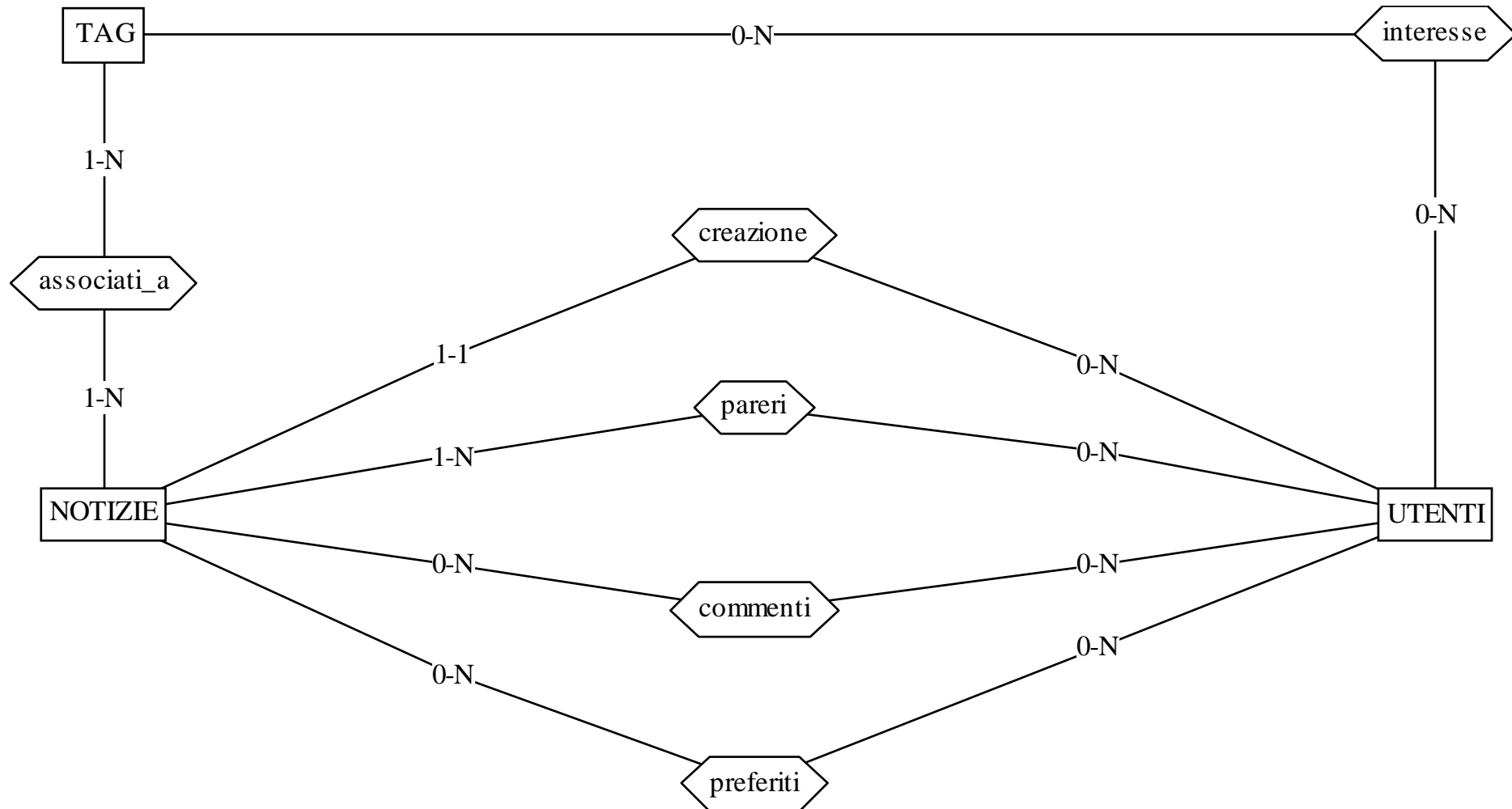
- Ogni **utente** può inserire delle **notizie**, indicando per ognuna un titolo e un link (esterno) di collegamento alla notizia completa. Eventualmente si può indicare anche una piccola frase che introduce la notizia. Gli utenti possono collaborare solo se sono registrati al sito (gratuitamente). Per ogni account si registra Username, Password, Nome, Cognome e email.
- Chi accede al sito legge le notizie e (registrandosi) può indicare un parere positivo o negativo (non più di uno per ogni notizia). Gli utenti registrati possono anche lasciare commenti alle notizie che leggono (anche più commenti per la stessa notizia), e indicare le proprie notizie preferite. Il sistema registra data e ora di inserimento per ogni notizia, per ogni voto e per ogni commento.
- Per facilitarne la gestione, ad ogni notizia vengono associate alcune parole chiave, o **“tag”** (almeno una) scelte dall'autore. Chi si dichiara interessato ad alcuni tag può ricevere periodicamente via email le notizie relative.



Modello E/R (1.2): associazioni

- Ogni **utente** può inserire delle **notizie**, indicando per ognuna un titolo e un link (esterno) di collegamento alla notizia completa. Eventualmente si può indicare anche una piccola frase che introduce la notizia. Gli utenti possono collaborare solo se sono registrati al sito (gratuitamente). Per ogni account si registra Username, Password, Nome, Cognome e email.
- Chi accede al sito legge le notizie e (registrandosi) può indicare un **parere** positivo o negativo (non più di uno per ogni notizia). Gli utenti registrati possono anche lasciare **commenti** alle notizie che leggono (anche più commenti per la stessa notizia), e indicare le proprie notizie **preferite**. Il sistema registra data e ora di inserimento per ogni notizia, per ogni voto e per ogni commento.
- Per facilitarne la gestione, ad ogni notizia vengono **associate** alcune parole chiave, o **"tag"** (almeno una) scelte dall'autore. Chi si dichiara **interessato** ad alcuni tag può ricevere periodicamente via email le notizie relative.

Modello E/R (1.3): schema di base



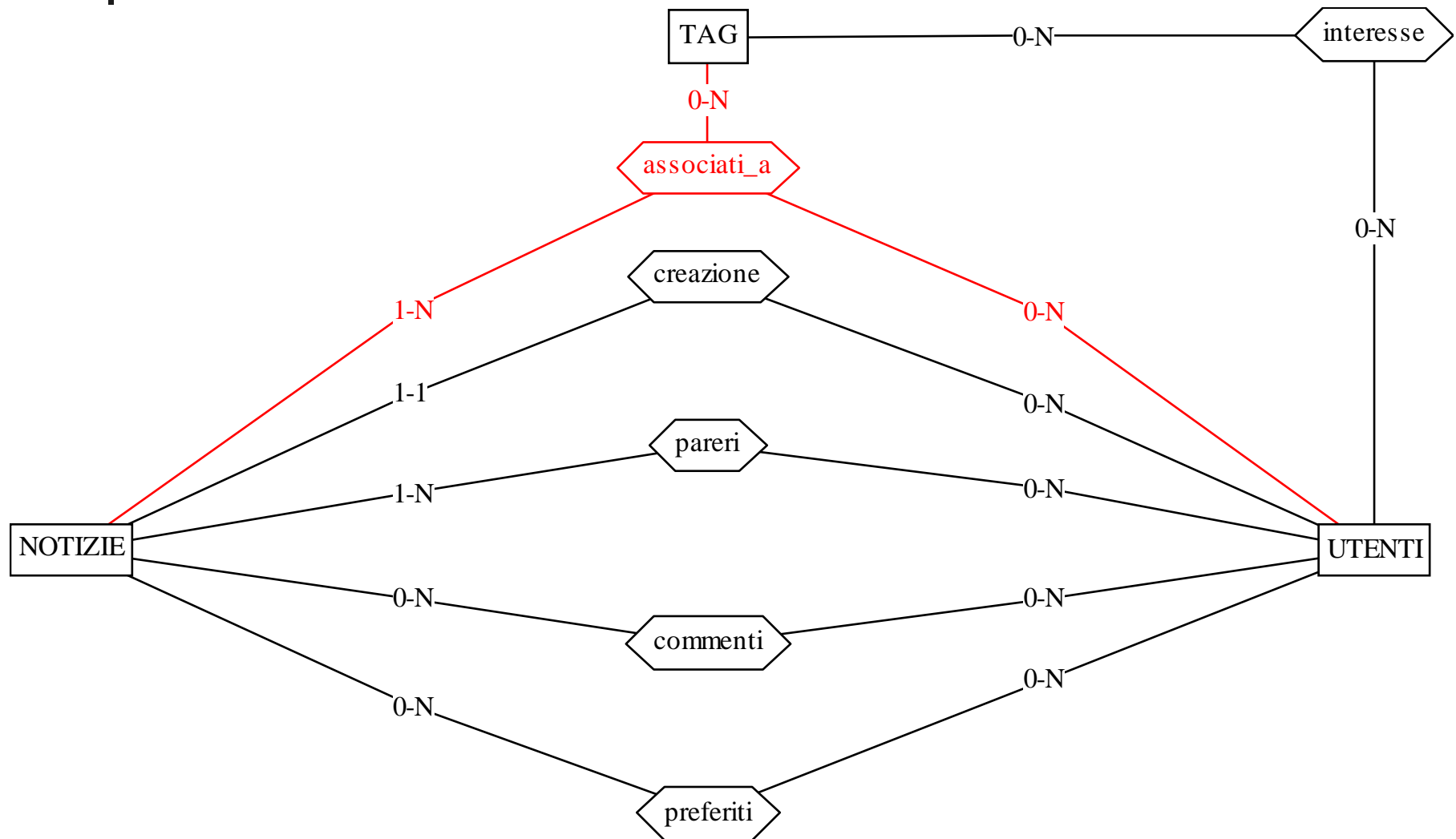


Modello E/R (1.4)

Si supponga ora di cambiare le specifiche come segue, in modo da riflettere uno scenario di “tagging collaborativo”:

- Per facilitarne la gestione, ad ogni notizia vengono associate alcune parole chiave, o “tag” (almeno una) scelte dall'autore **o anche da altri utenti...**
- **Come si modifica lo schema visto?**

Modello E/R (1.5): schema modificato



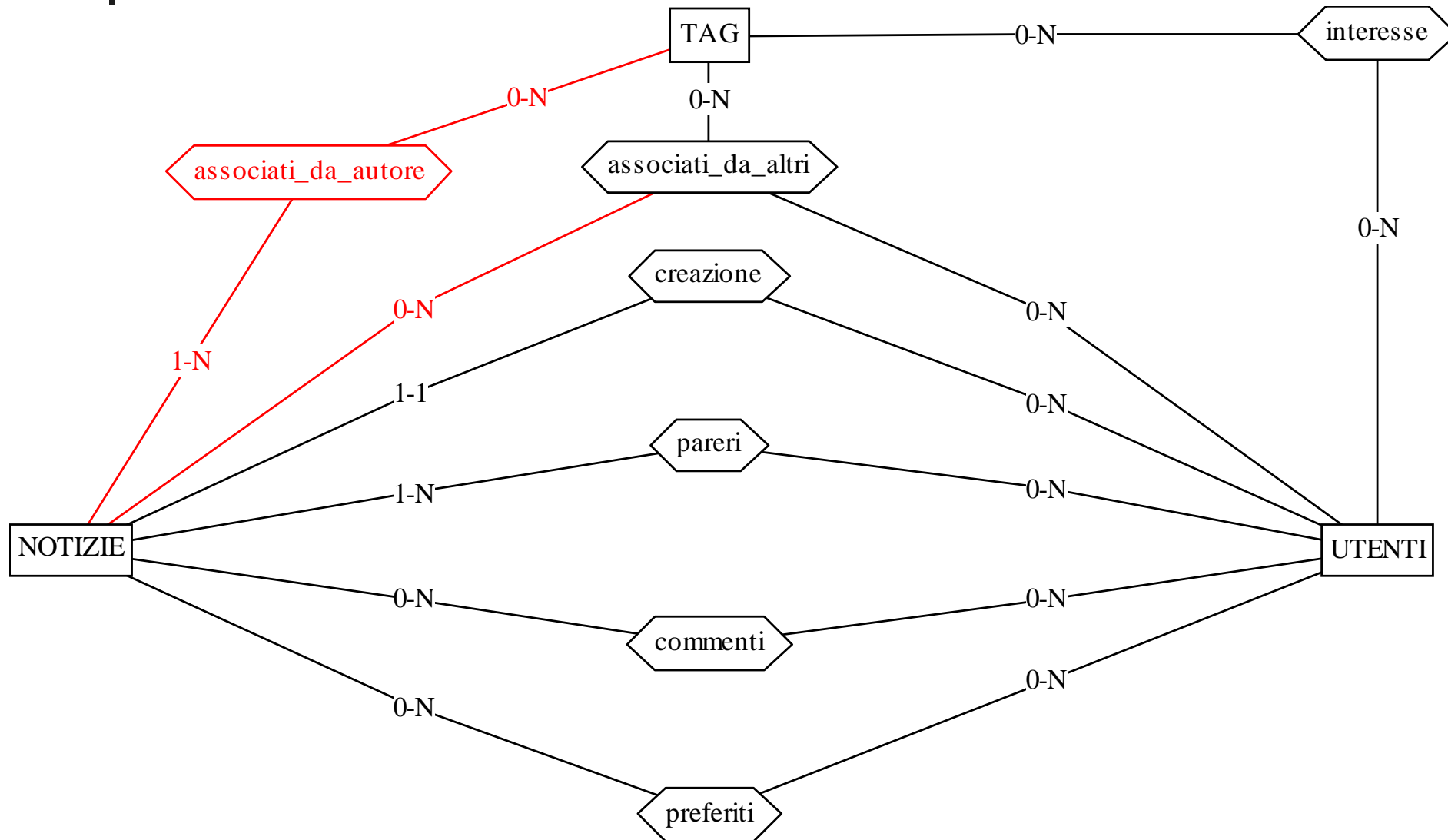


Modello E/R (1.6)

Si supponga ora che le specifiche siano estese come segue:

- Per facilitarne la gestione, ad ogni notizia vengono associate alcune parole chiave, o “tag” (almeno una) scelte dall’autore o anche da altri utenti, **avendo cura di mantenere distinti i tag assegnati dall’autore dagli altri**
- **Come si modifica lo schema precedente?**

Modello E/R (1.7): schema modificato





Modello E/R (1.8)

Considerazioni:

- L' ultimo schema si ottiene dal precedente partizionando le istanze dell'associazione **associati_a**, **NON** le istanze dell'entità **TAG**
- Lo schema E/R derivato **NON** esprime il vincolo che all'associazione **associati_da_altri** non può partecipare l'autore della notizia, ovvero:

non possono esistere una tripla $(t, n, u) \in \text{associati_da_altri}$ e una coppia $(n, u) \in \text{creazione}$ in cui l'utente **u** è lo stesso

- Vincoli di questo tipo non sono esprimibili mediante il modello E/R



Modello E/R (2.1)

Progettare uno schema E/R per le seguenti specifiche:

- “C’è un Fiore per Te...” è un’organizzazione on-line in grado di recapitare composizioni floreali di vario genere 24 ore su 24 sia in territorio nazionale che europeo. Il cliente interessato, specificando semplicemente l’indirizzo di destinazione (via, località e comune) unitamente a nome e cognome della persona “fortunata” e a un messaggio personalizzato, può ottenere, per esempio, la spedizione in giornata di uno splendido mazzo di tulipani frastagliati. In caso di spedizione in Italia occorre indicare anche il CAP della località, mentre per consegne fuori Italia occorre specificare anche il nome del paese di destinazione.
- L’elevata qualità del servizio offerto dipende dal fatto che in ogni località, nazionale o europea, l’organizzazione si avvale di diversi fornitori (un fornitore in generale copre diverse località). Per un determinato prodotto floreale, e in funzione del livello di qualità scelto dal cliente (base, “per chi vuole lasciare il segno”, superiore alla media e “per chi vuole il massimo”), ogni fornitore ha un proprio prezzo e un tempo massimo garantito di consegna. Pertanto l’utente può scegliere il fornitore più conveniente per la propria consegna.
- La gamma di prodotti messi a disposizione del cliente comprende mazzi preconfezionati di fiori di diversa tipologia (rose, tulipani, orchidee...), composizioni floreali in ciotole di specifica dimensione, piante (di specifica altezza) e centrotavola caratterizzati da uno specifico numero di candele. Al termine dell’ordine (si noti che un ordine può comprendere diversi prodotti floreali indirizzati a diverse persone), il cliente effettua il pagamento dell’importo totale specificando il numero di carta di credito.



Modello E/R (2.2)

Specifiche emendate:

- “C’è un Fiore per Te...” è un’organizzazione on-line in grado di recapitare composizioni floreali di vario genere 24 ore su 24 sia in territorio nazionale che europeo. Il cliente interessato, specificando semplicemente l’indirizzo di destinazione (via, località e comune) unitamente a nome e cognome della persona “fortunata” e a un messaggio personalizzato, può ottenere, per esempio, la spedizione in giornata di uno splendido mazzo di tulipani frastagliati. In caso di spedizione in Italia occorre indicare anche il CAP della località, mentre per consegne fuori Italia occorre specificare anche il nome del paese di destinazione.
- L’elevata qualità del servizio offerto dipende dal fatto che in ogni località, nazionale o europea, l’organizzazione si avvale di diversi fornitori (un fornitore in generale copre diverse località). Per un determinato prodotto floreale, e in funzione del livello di qualità scelto dal cliente (base, “per chi vuole lasciare il segno”, superiore alla media e “per chi vuole il massimo”), ogni fornitore ha un proprio prezzo e un tempo massimo garantito di consegna. Pertanto l’utente può scegliere il fornitore più conveniente per la propria consegna.
- La gamma di prodotti messi a disposizione del cliente comprende mazzi preconfezionati di fiori di diversa tipologia (rose, tulipani, orchidee...), composizioni floreali in ciotole di specifica dimensione, piante (di specifica altezza) e centrotavola caratterizzati da uno specifico numero di candele. Al termine dell’ordine (si noti che un ordine può comprendere diversi prodotti floreali indirizzati a diverse persone), il cliente effettua il pagamento dell’importo totale specificando il numero di carta di credito.



Modello E/R (2.3)

Specifiche riorganizzate:

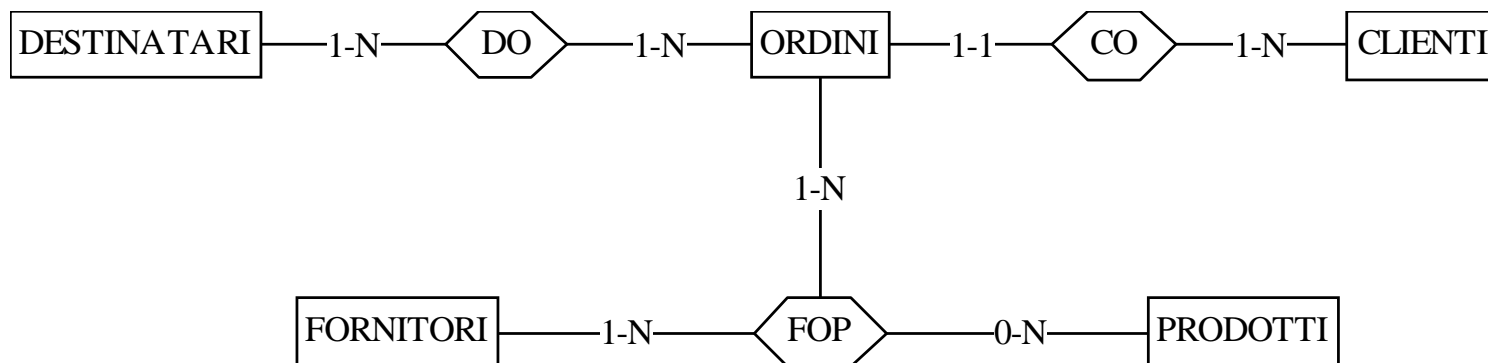
- Un destinatario è caratterizzato dall'indirizzo di destinazione (via, località e comune) e dal nome e cognome. Se in Italia, anche dal CAP, all'estero anche dal paese.
 - Per ogni invio si prevede anche un messaggio personalizzato.
- In ogni località vi sono diversi fornitori (un fornitore può coprire diverse località). Per un determinato prodotto floreale, e in funzione del livello di qualità scelto dal cliente, ogni fornitore ha un proprio prezzo e un tempo massimo garantito di consegna.
- I prodotti comprendono mazzi preconfezionati di fiori di diversa tipologia, composizioni floreali in ciotole di specifica dimensione, piante (di specifica altezza) e centrotavola caratterizzati da uno specifico numero di candele. Un ordine può comprendere diversi prodotti floreali indirizzati a diverse persone. Per il pagamento dell'importo totale serve il numero di carta di credito.

Modello E/R (2.4)

Schema scheletro (1):

- Sulla base delle specifiche si possono innanzitutto derivare le entità:
 - Destinatari
 - Fornitori
 - Clienti
 - Prodotti
 - Ordini

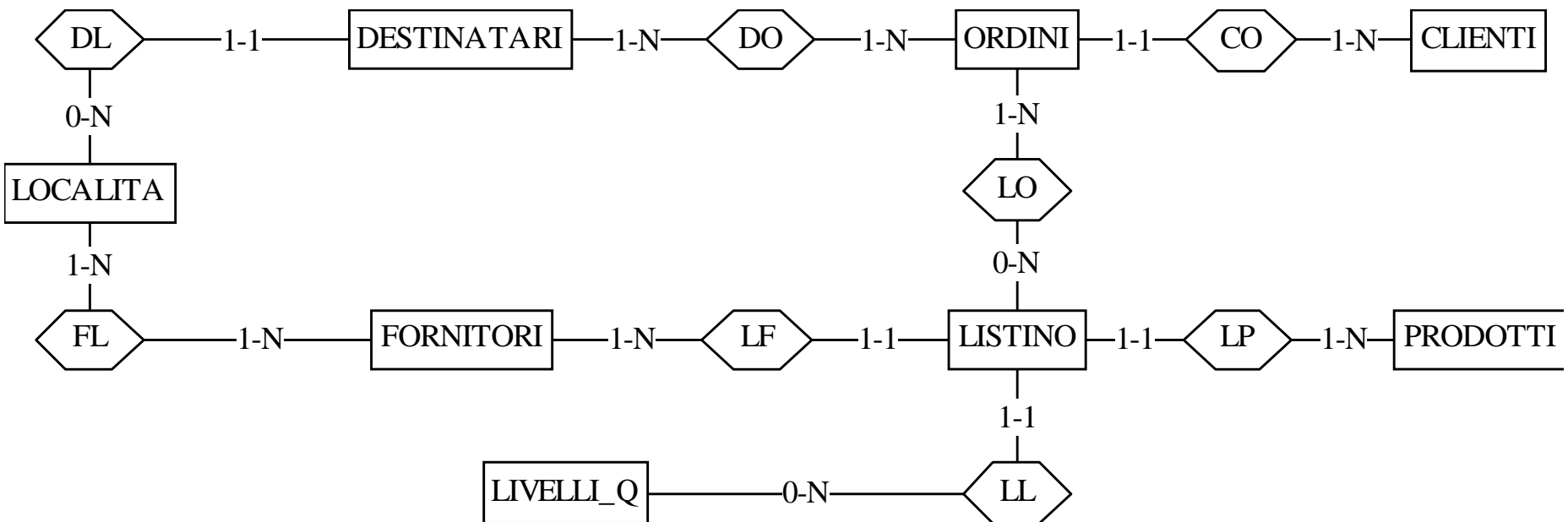
che danno luogo a una prima versione dello schema scheletro, in cui l'associazione ternaria FOP modella il requisito che un ordine possa riguardare diversi prodotti e diversi fornitori



Modello E/R (2.5)

Schema scheletro (2):

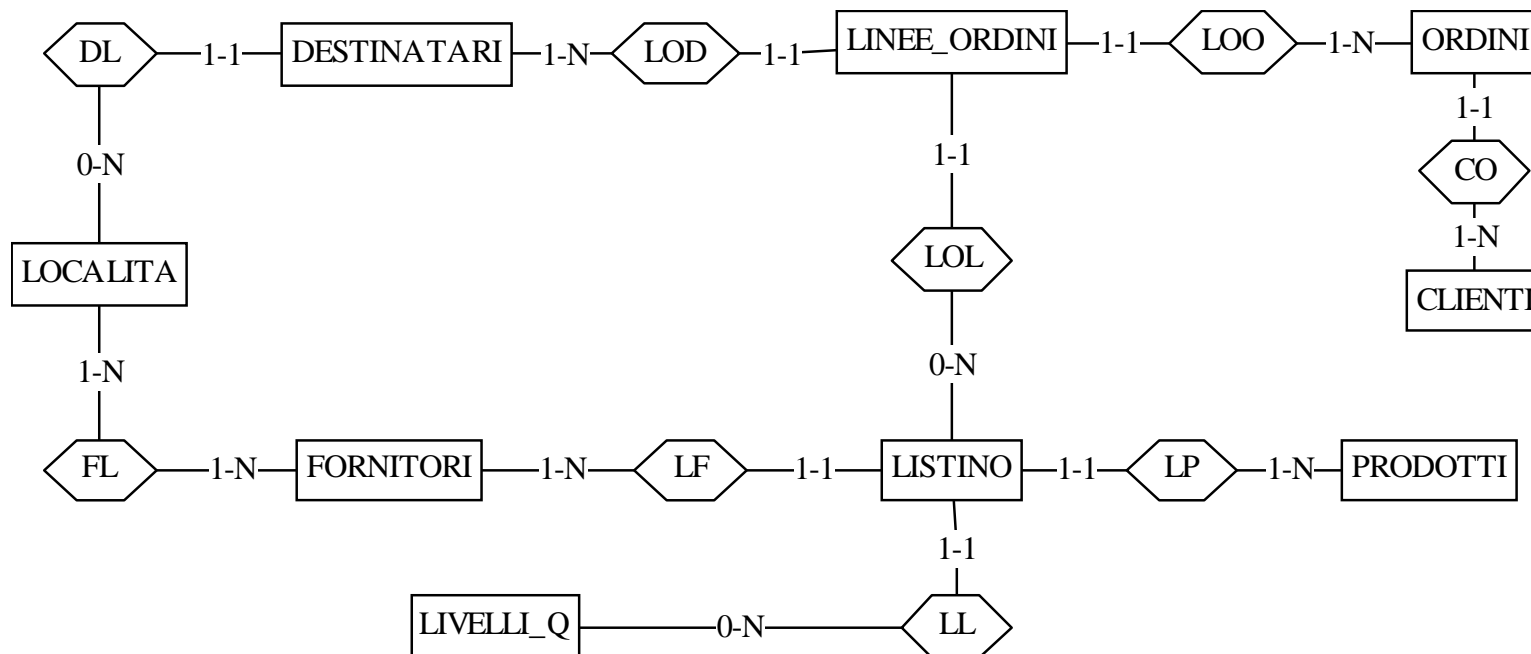
- Lo schema scheletro si raffina introducendo la nuova entità **Località**, che si relaziona a Destinatarî e a Fornitori
- Appare inoltre opportuno “reificare” l’associazione FOP in una nuova entità (**Listino**) il cui scopo è mantenere informazioni “statiche” (e quindi indipendenti dagli Ordini) su prezzi e tempi massimi di consegna. Ciò comporta anche l’introduzione di una nuova entità **LivelliQ** (livelli di qualità), da cui tali informazioni dipendono



Modello E/R (2.6)

Derivazione dello Schema Concettuale (1):

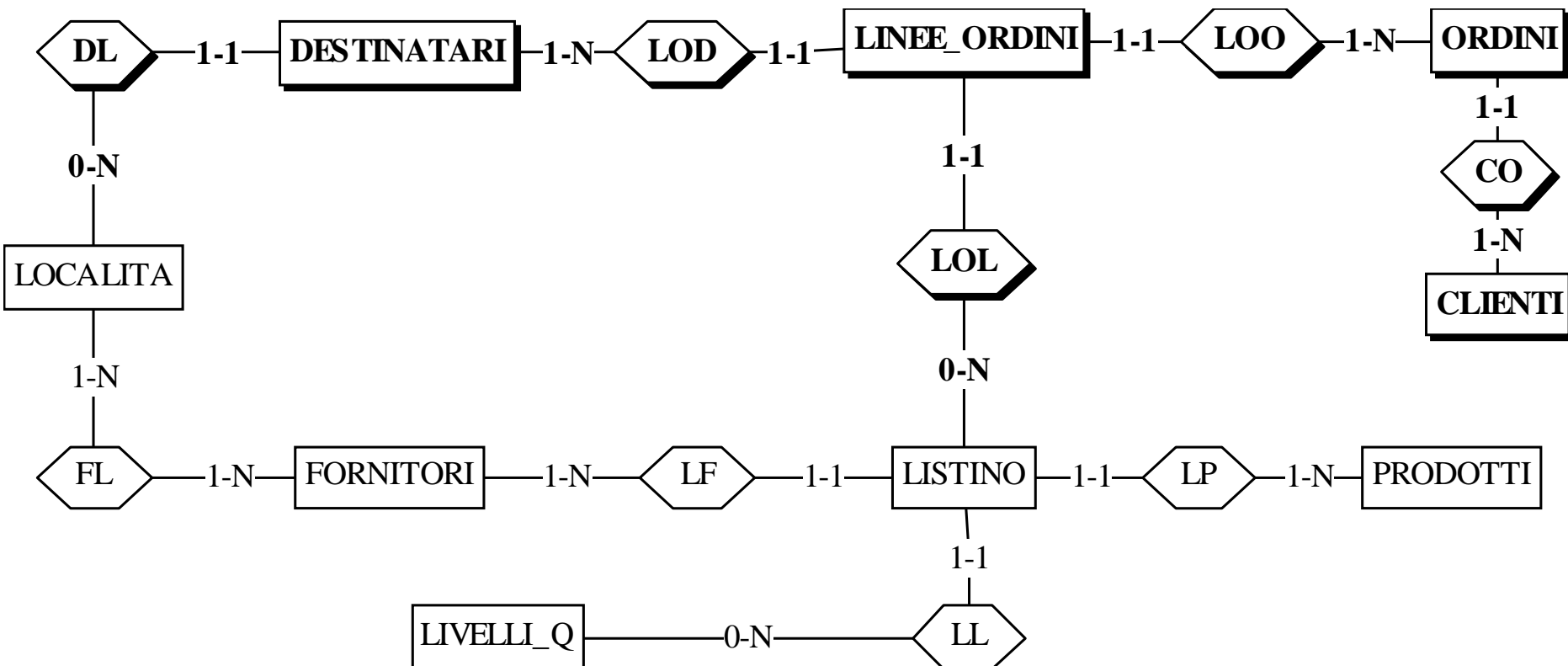
- Un primo raffinamento dello schema scheletro può consistere nella modellazione più precisa degli Ordini, allo scopo di relazionare i singoli elementi che costituiscono un ordine con i rispettivi destinatari. Sono possibili 2 strategie:
 - Trasformare l'associazione LO in ternaria, aggiungendo un ramo verso Destinatari, e rimuovere conseguentemente l'associazione DO
 - Reificare LO in LineeOrdini
- La prima alternativa non supporta naturalmente il caso di Ordini con elementi di listino ripetuti e relativi allo stesso destinatario. Si opta quindi per la seconda alternativa



Modello E/R (2.7)

Commenti:

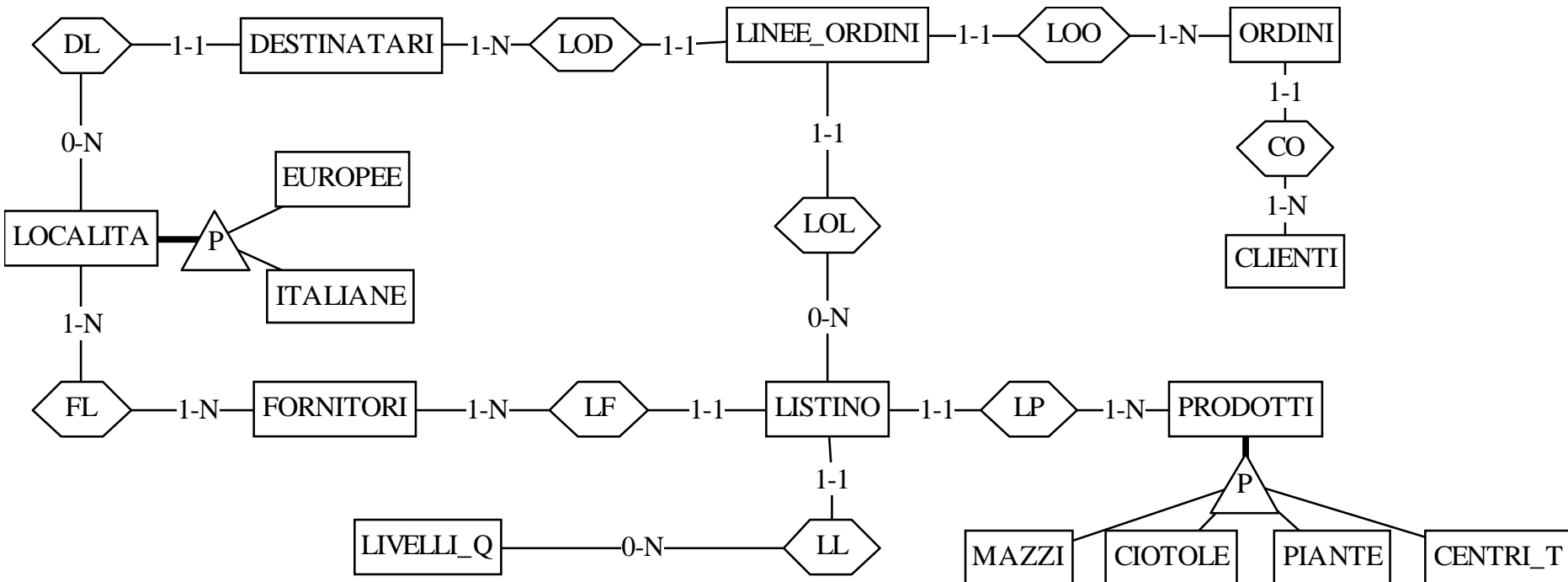
- E' opportuno notare la chiara distinzione tra la parte "statica" e quella più propriamente "dinamica" del DB, quest'ultima evidenziata in rilievo



Modello E/R (2.8)

Derivazione dello Schema Concettuale (2):

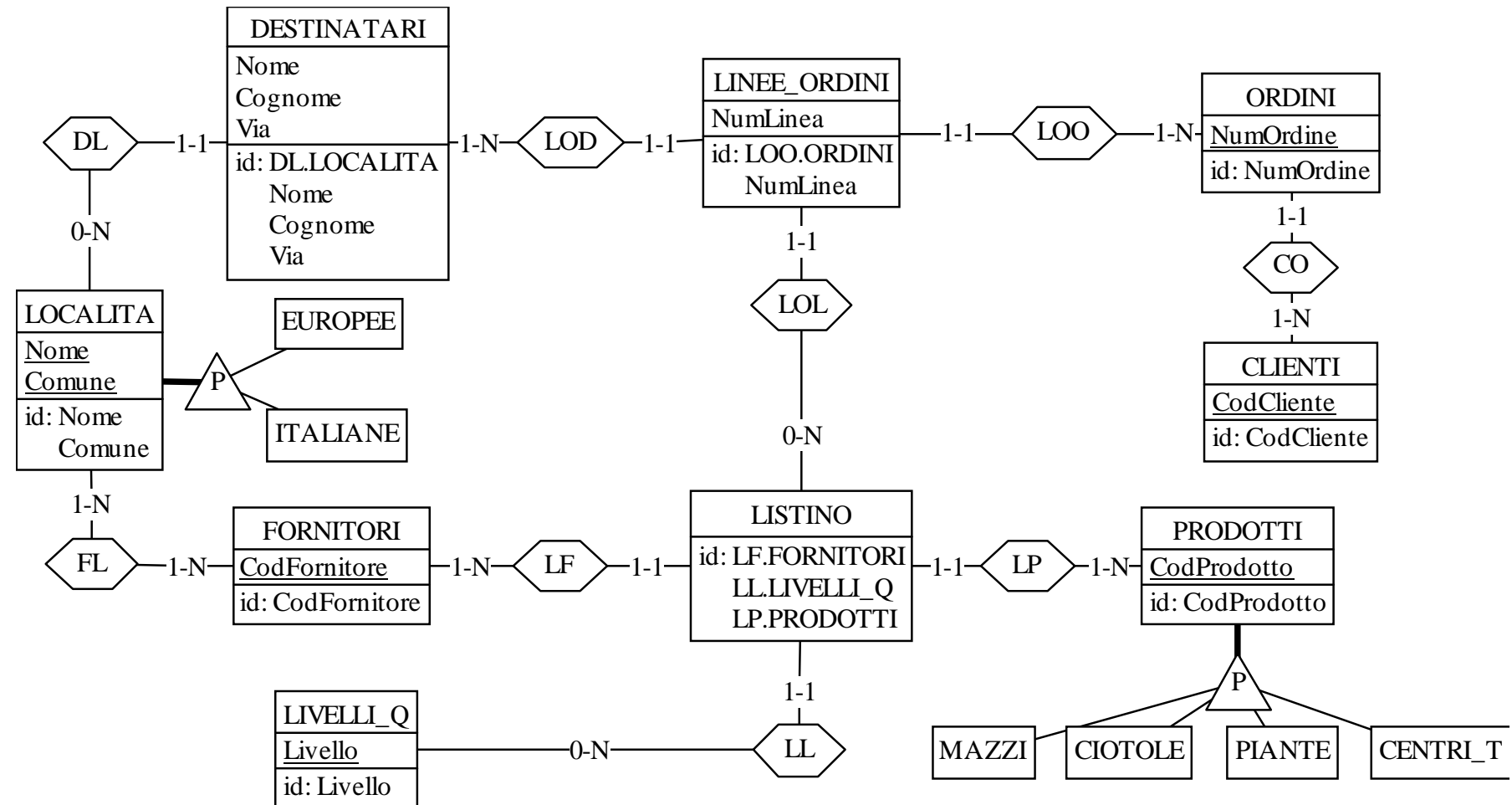
- Si procede quindi con il raffinamento delle entità Località e Prodotti:



Modello E/R (2.9)

Derivazione dello Schema Concettuale (3):

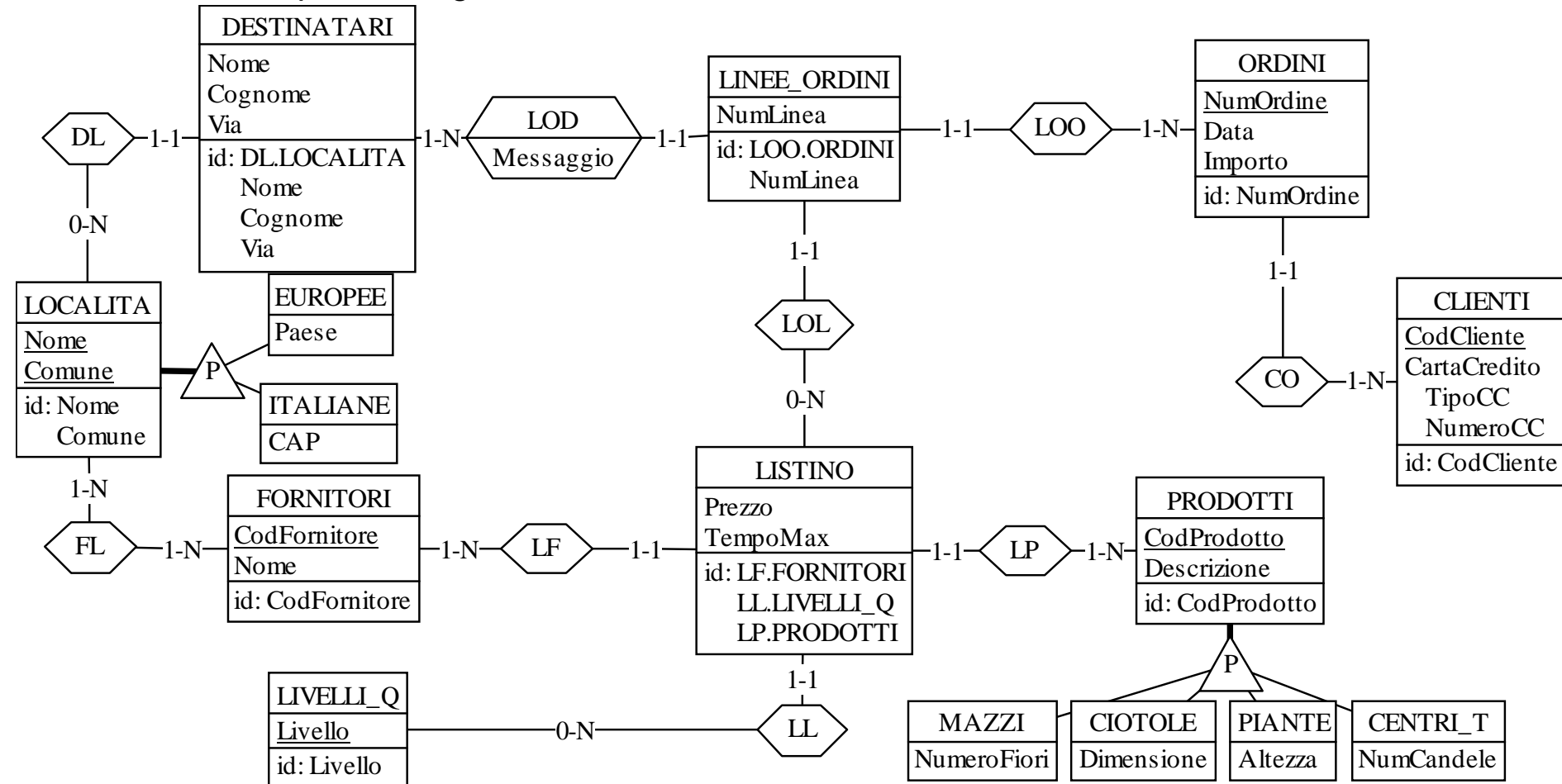
- Si definiscono gli identificatori per tutte le entità:



Modello E/R (2.10)

Derivazione dello Schema Concettuale (4):

- Infine si completa con gli altri attributi di entità e associazioni:





Modello E/R (2.11)

Commenti:

- L'attributo Descrizione di Prodotti permette di descrivere sia, ad es., il tipo di fiore per i mazzi che la varietà di pianta, ecc.
- Ovviamente per un destinatario è anche possibile specificare più informazioni (numero civico, telefono, ecc.)
- Idem per Clienti e Fornitori
- E' anche ragionevole prevedere la possibilità di più carte di credito per uno stesso cliente

Vincoli non espressi dallo schema:

- L'attributo Ordini.Importo è pari alla somma dei prezzi di listino degli elementi acquistati
- Il fornitore scelto per una consegna copre la località del destinatario



Modello E/R (2.12)

Controllo di completezza:

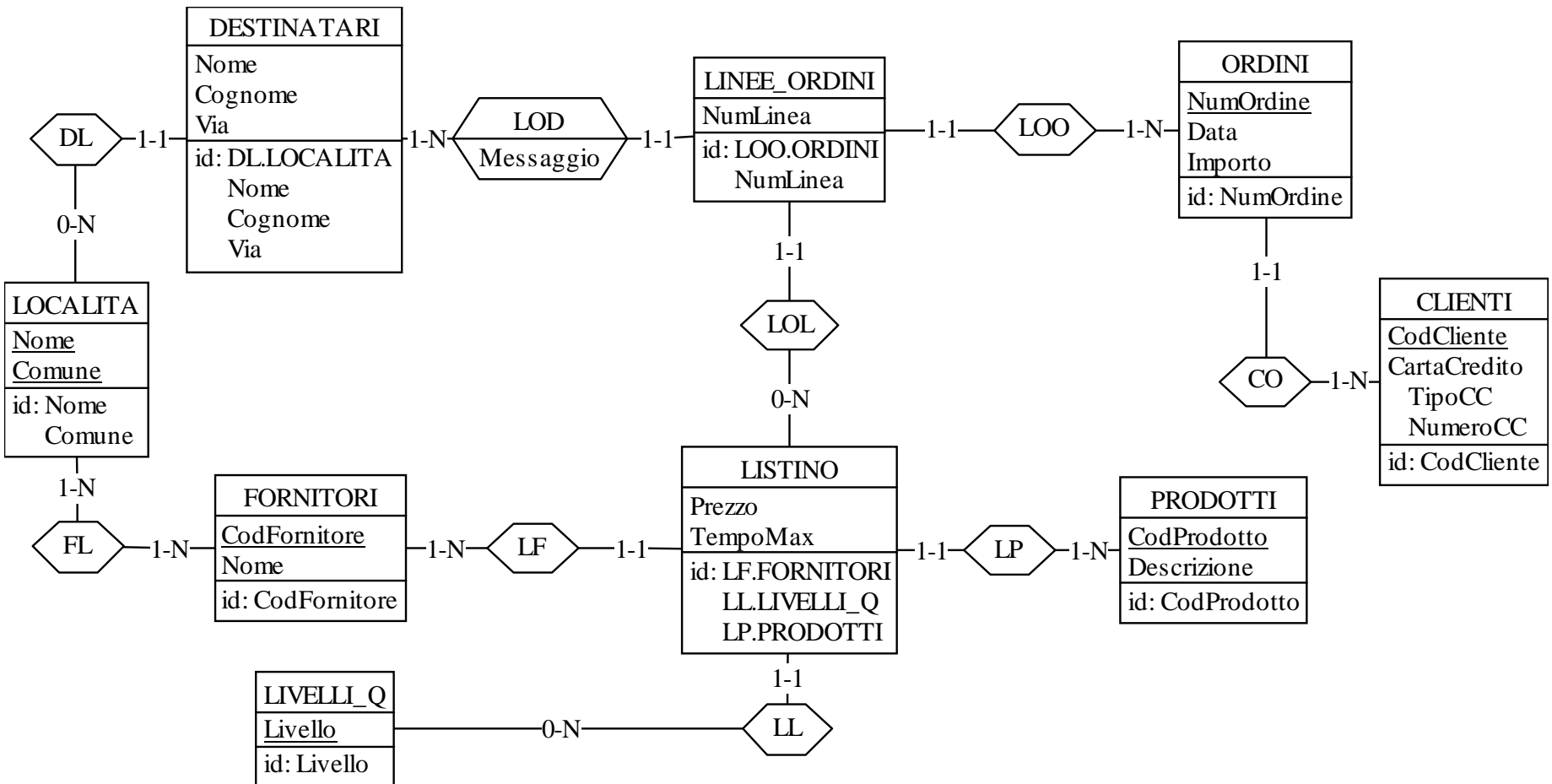
- Un destinatario è caratterizzato dall'indirizzo di destinazione (via, località e comune) e dal nome e cognome. Se in Italia, anche dal CAP, all'estero anche dal paese. **OK**
 - Per ogni invio si prevede anche un messaggio personalizzato. **OK**
- In ogni località vi sono diversi fornitori (un fornitore può coprire diverse località). **OK**

Per un determinato prodotto floreale, e in funzione del livello di qualità scelto dal cliente, ogni fornitore ha un proprio prezzo e un tempo massimo garantito di consegna. **OK**
- I prodotti comprendono mazzi preconfezionati di fiori di diversa tipologia, composizioni floreali in ciotole di specifica dimensione, piante (di specifica altezza) e centrotavola con uno specifico numero di candele. **OK**

Un ordine può comprendere diversi prodotti floreali indirizzati a diverse persone. **OK**
Per il pagamento dell'importo totale serve il numero di carta di credito. **OK**

Progettazione logica (1.1)

Progettare uno schema relazionale per lo schema E/R derivato, ignorando in prima istanza le entità specializzate di Prodotti e Località, ovvero:





Progettazione logica (1.2)

Soluzione:

- La traduzione non presenta particolari problemi, anche perché le cardinalità minime delle associazioni che si possono tradurre insieme a un'entità sono tutte pari a 1. Si ha pertanto:

PRODOTTI(CodProdotto, Descrizione)

FORNITORI(CodFornitore, Nome)

LIVELLI_QUALITA(Livello)

LISTINO(CodProdotto, CodFornitore, Livello, Prezzo, TempoMax) FK:...

LOCALITA(Nome, Comune)

COPERTURE(CodFornitore, NomeLocalità, Comune) FK: ...

CLIENTI(CodCliente, TipoCartaCredito, NumeroCartaCredito)

ORDINI(NumOrdine, Data, Importo, CodCliente) FK: ...

DESTINATARI(Nome, Cognome, Via, NomeLocalità, Comune) FK: ...

LINEE_ORDINI(NumOrdine, NumLinea, CodProdotto, CodFornitore, Livello, Messaggio,
NomeDestinatario, CognomeDestinatario, Via, NomeLocalità, Comune) FK: ...



Progettazione logica (1.3)

Commenti e raffinamenti:

- Nella soluzione presentata è evidente la “pesantezza” di alcune foreign key. Se si decide di introdurre dei codici ad hoc bisogna tuttavia essere coscienti del fatto che ciò può comportare join aggiuntivi in fase di ricerca (ad es.: dati del destinatario di una linea d'ordine). Si ottiene:

PRODOTTI(CodProdotto,Descrizione)

FORNITORI(CodFornitore,Nome)

LIVELLI_QUALITA(Livello)

LISTINO(CodListino,CodProdotto,CodFornitore,Livello,Prezzo,TempoMax) UNIQUE(...)

LOCALITA(Nome,Comune)

COPERTURE(CodFornitore, NomeLocalità,Comune) FK: ...

CLIENTI(CodCliente,TipoCartaCredito,NumeroCartaCredito)

ORDINI(NumOrdine,Data,Importo,CodCliente) FK: ...

DESTINATARI(CodDestinatario,Nome,Cognome,Via,NomeLocalità,Comune) UNIQUE(...)

LINEE_ORDINI(NumOrdine,NumLinea,CodListino,CodDestinatario,Messaggio)

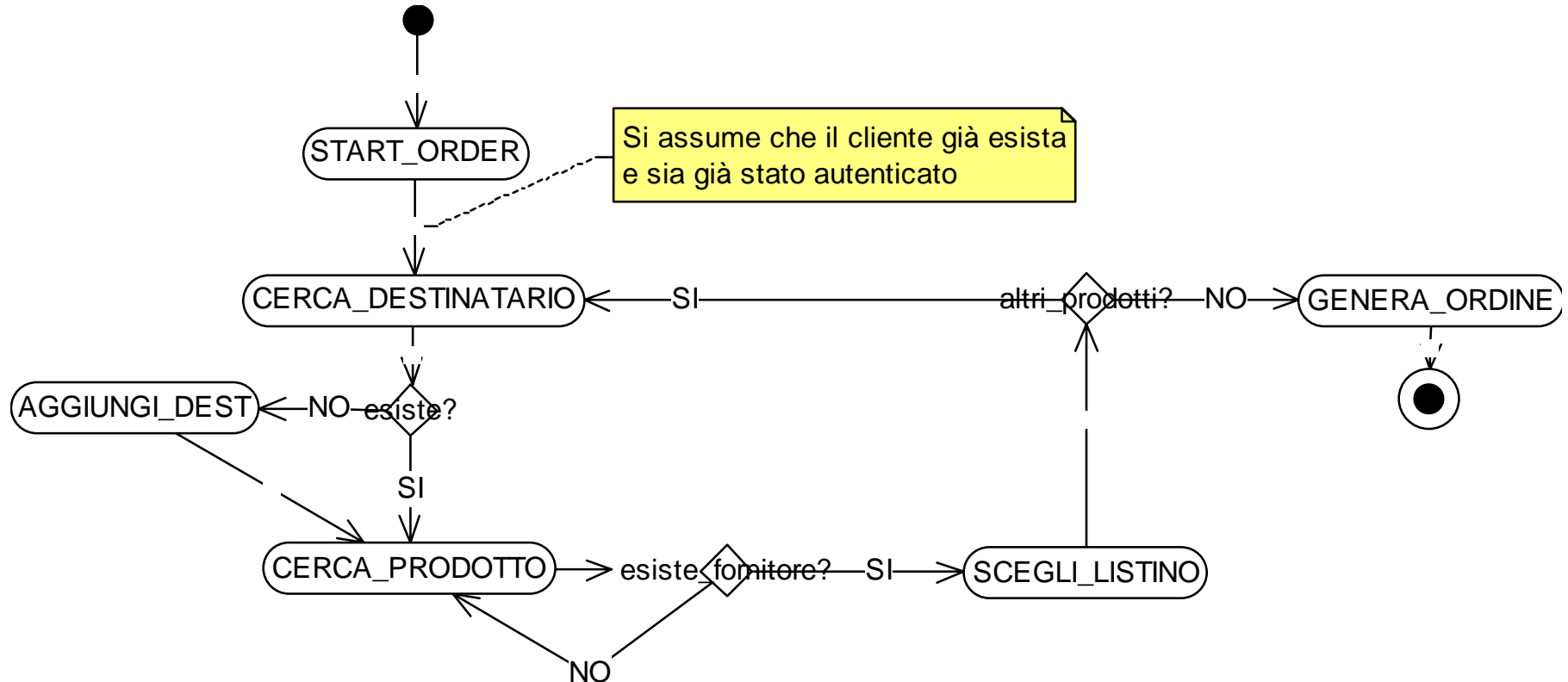
Progettazione logica (1.4)

Esempio di operazione:

- A titolo esemplificativo, si analizza come verrebbe ad essere eseguita, con riferimento all'ultimo DB generato, l'operazione di

INSERIMENTO DI UN NUOVO ORDINE

una cui possibile logica (semplificata) si riassume come in figura:





Progettazione logica (1.5)

CERCA_DESTINATARIO:

- Le modalità possibili di ricerca sono varie, nel caso si specifichi solo nome e cognome:

```
SELECT *  
FROM DESTINATARI  
WHERE Cognome = :cognome  
AND Nome = :nome
```

- Se il destinatario esiste viene selezionato dal cliente, altrimenti

AGGIUNGI_DESTINATARIO:

- Il cliente inserisce tutte le informazioni del nuovo destinatario:

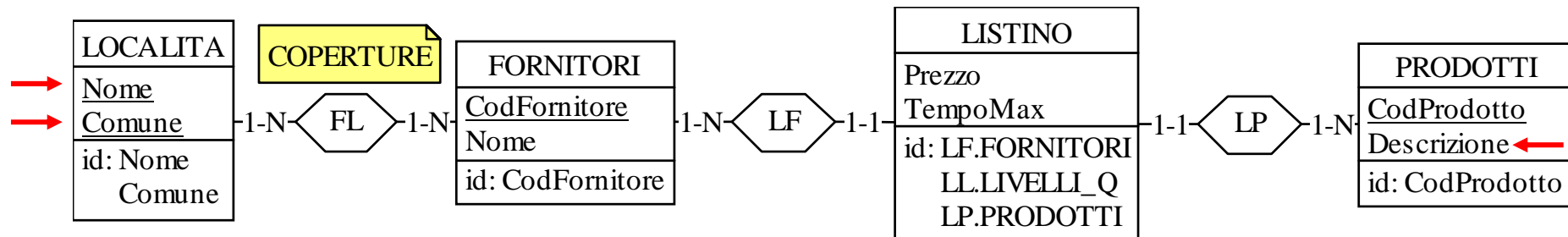
```
INSERT INTO DESTINATARI(Nome,Cognome,Via,NomeLocalità,Comune)  
VALUES (:nome,:cognome,:via,:località,:comune)
```

e il sistema provvede a generare il codice corrispondente

Progettazione logica (1.6)

CERCA_PRODOTTO:

- L'utente ricerca un prodotto, ad es. specificando una stringa di descrizione (variabile **:descrizione**), e il sistema verifica che esista almeno un fornitore per la località scelta:



```
SELECT *  
FROM PRODOTTI P, LISTINO L, COPERTURE C  
WHERE P.Descrizione LIKE '%' CONCAT :descrizione CONCAT '%'  
AND P.CodProdotto = L.CodProdotto  
AND L.CodFornitore = C.CodFornitore  
AND C.NomeLocalità = :località  
AND C.Comune = :comune
```

- Se la query non restituisce alcun risultato viene richiesta una nuova ricerca di prodotti



Progettazione logica (1.7)

SCEGLI_LISTINO:

- Nel caso di più alternative, l'utente ne sceglie comunque una, scrive il messaggio e il sistema produce la corrispondente linea d'ordine (generando anche i valori per **NumOrdine** e **NumLinea**):

```
INSERT INTO LINEE_ORDINI(CodListino,CodDestinatario,Messaggio)
VALUES (:codlistino,:coddestinatario,:messaggio)
```

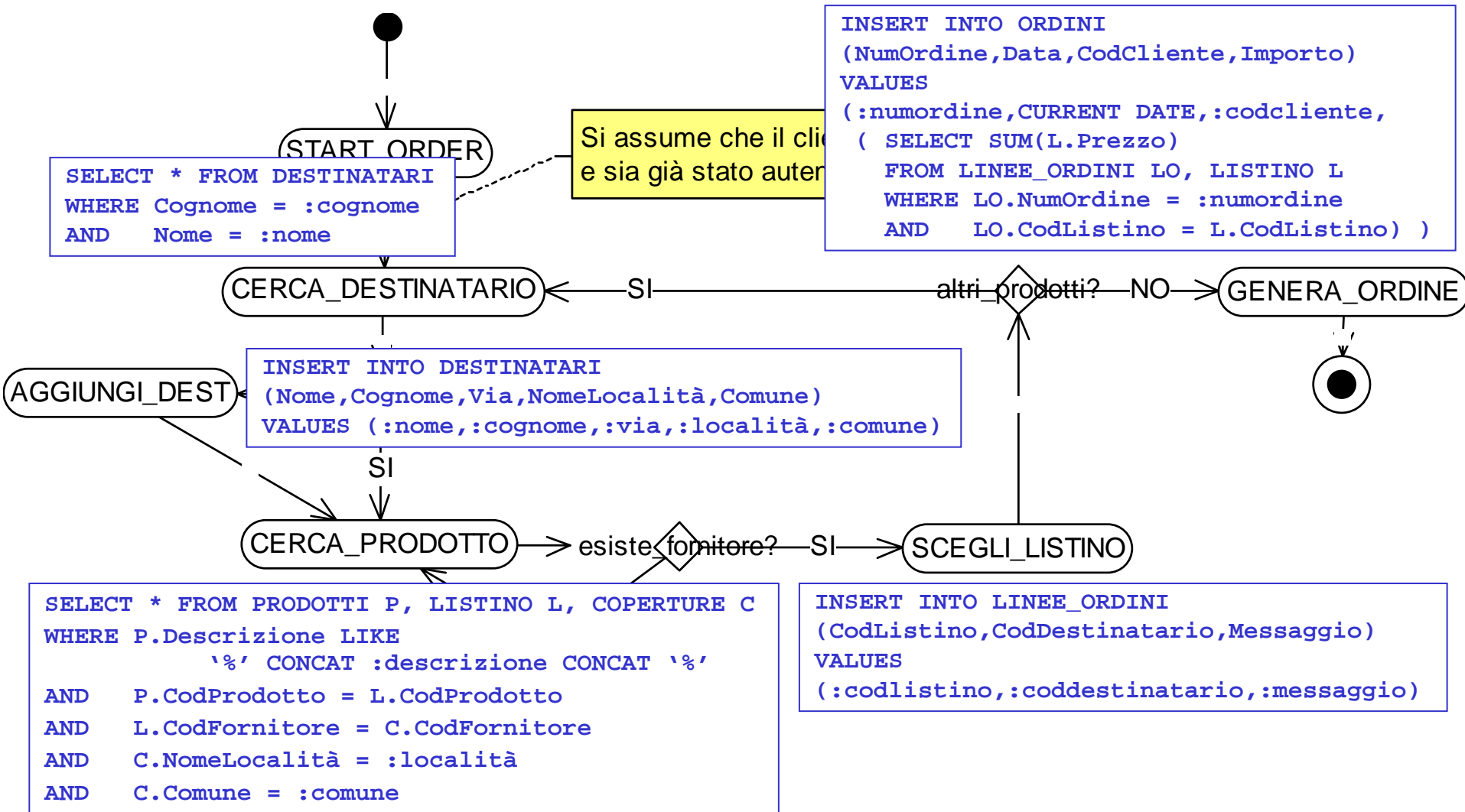
GENERA_ORDINE:

- Quando tutte le linee d'ordine sono state generate, il sistema calcola l'importo e genera il nuovo ordine:

```
INSERT INTO ORDINI(NumOrdine,Data,CodCliente,Importo)
VALUES (:numordine,CURRENT DATE,:codcliente,
        ( SELECT SUM(L.Prezzo)
          FROM LINEE_ORDINI LO, LISTINO L
          WHERE LO.NumOrdine = :numordine
                AND LO.CodListino = L.CodListino) )
```

- Ovviamente è possibile evitare di eseguire una query aggregata se l'applicazione mantiene in memoria le informazioni sugli elementi dell'ordine

Progettazione logica (1.8)





Progettazione logica (2.1)

RISPETTO DEI VINCOLI:

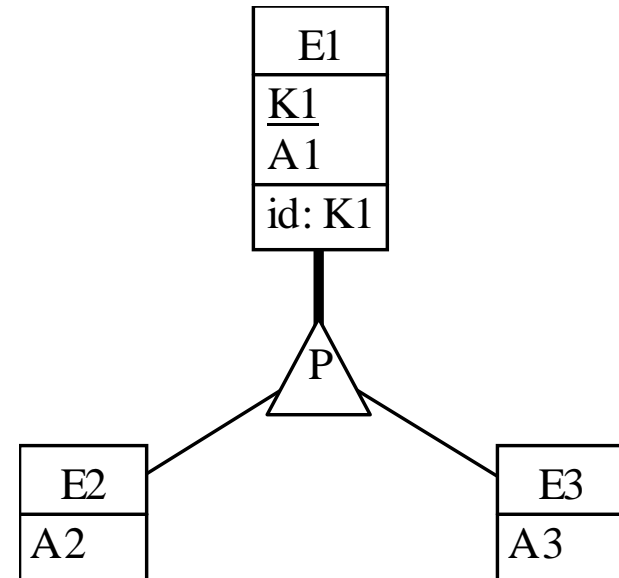
- La traduzione da uno schema E/R a uno schema relazionale non sempre consente di definire schemi di relazioni in grado di imporre tutti i vincoli presenti a livello concettuale. Ciò è vero in particolare nel caso di traduzione di gerarchie.
- Inoltre, va anche considerato il caso di vincoli che nemmeno a livello concettuale erano formalmente esprimibili
- Questa serie di esercizi di progettazione logica pone l'enfasi sul primo di questi aspetti

Progettazione logica (2.2)

GERARCHIE (T,E) = P in DB-Main:

- Si consideri la traduzione della seguente gerarchia (supponendo per semplicità che tutti gli attributi dello schema siano di tipo INT)

mediante 2 relazioni (collasso verso il basso):



```
Create table E2 (  
  K1 int not null primary key,  
  A1 int not null,  
  A2 int not null );
```

```
Create table E3 (  
  K1 int not null primary key,  
  A1 int not null,  
  A3 int not null );
```


Progettazione logica (2.3)

Vincoli aggiuntivi:

- Occorre garantire che le istanze di E2 ed E3 siano effettivamente disgiunte
- Se DB2 consentisse clausole CHECK arbitrariamente complesse si potrebbe aggiungere allo schema di E2 il vincolo (analogamente per E3):

```
Create table E2 (  
    K1 int not null primary key,  
    A1 int not null,  
    A2 int not null,  
    constraint exclusive_hierarchy  
        check (not exists (select * from E3 where E3.K1 = K1)) );
```

```
Create table E3 (  
    K1 int not null primary key,  
    A1 int not null,  
    A3 int not null,  
    constraint exclusive_hierarchy  
        check (not exists (select * from E2 where E2.K1 = K1)) );
```



Progettazione logica (2.4)

Sopperire ai limiti della clausola CHECK mediante query SQL:

- Il vincolo **exclusive_hierarchy** può essere imposto a livello di applicazione mediante un'equivalente *query di verifica*, da eseguire prima di inserire una tupla in E2 (analogamente per E3)
- O, meglio, mediante **trigger**

```
-- OK se non restituisce nessuna tupla (not exists)
Select * from E3 where E3.K1 = :K1;
```

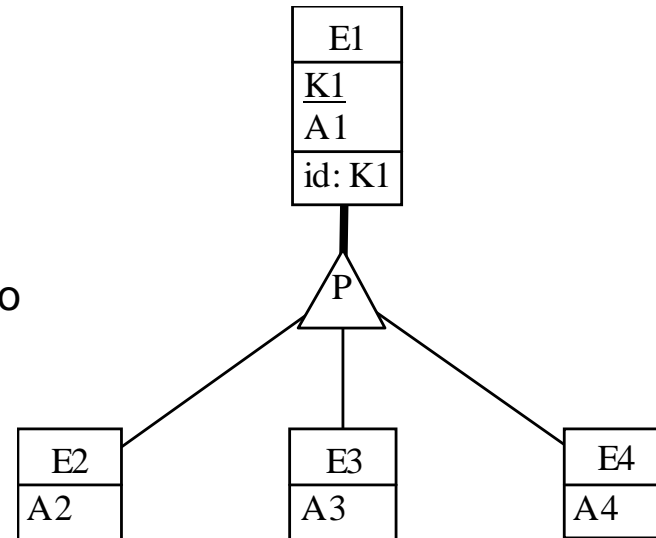
```
Insert into E2 (K1,A1,A2)
Values (:K1,:A1,:A2);
```

Progettazione logica (2.5)

Tre o più entità specializzate:

- Con uno schema del tipo:

bisogna garantire la mutua esclusione rispetto a tutte le altre entità figlie. Ad es. per E2:



-- OK se non restituisce nessuna tupla

```
Select E3.K1 from E3 where E3.K1 = :K1 -- NB: non Select *
```

UNION

```
Select E4.K1 from E4 where E4.K1 = :K1 ;
```

```
Insert into E2 (K1,A2)
```

```
Values (:K1,:A1,:A2);
```

Progettazione logica (2.6)

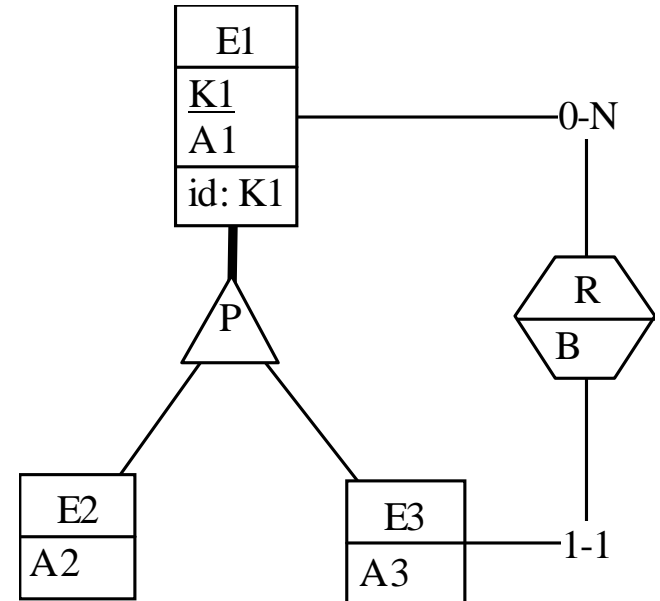
Collasso verso il basso con associazione:

- Lo schema:

viene tradotto collassando verso il basso e inglobando l'associazione R in E3:

```
Create table E2 (  
  K1 int not null primary key,  
  A1 int not null,  
  A2 int not null );
```

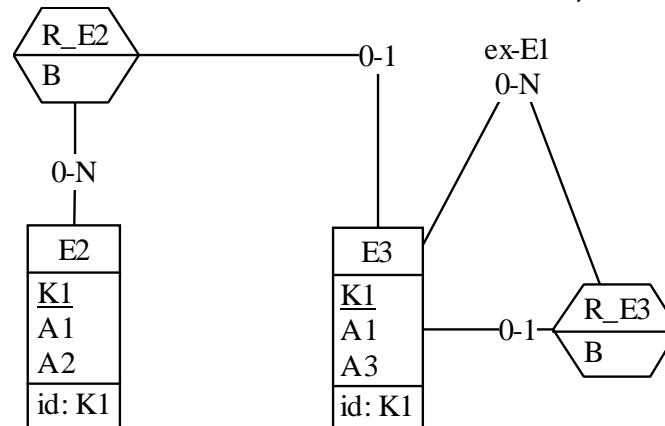
```
Create table E3 (  
  K1 int not null primary key,  
  A1 int not null,  
  A3 int not null ); -- manca l'associazione
```



Progettazione logica (2.7)

Traduzione dell'associazione:

- Per tradurre correttamente l'associazione, occorre tenere presente che un elemento di E3 ora può referenziare o un elemento di E3 o uno di E2, come se lo schema fosse:



- Si introducono quindi 2 foreign key, che possono anche assumere valore nullo:

```
Create table E3 (  
  K1 int not null primary key,  
  A1 int not null,  
  A3 int not null,  
  K1_R_E2 int references E2,  
  K1_R_E3 int references E3,  
  B int not null );
```



Progettazione logica (2.8)

Vincolo sull'associazione:

- Esattamente una delle due foreign key deve assumere valore nullo:

```
Create table E3 (  
  K1 int not null primary key,  
  A1 int not null,  
  A3 int not null,  
  K1_R_E2 int references E2,  
  K1_R_E3 int references E3,  
  B int not null,  
  constraint R_is_always_defined  
    check ( (K1_R_E2 is null and K1_R_E3 is not null) or  
            (K1_R_E3 is null and K1_R_E2 is not null) ) );
```

Progettazione logica (2.9)

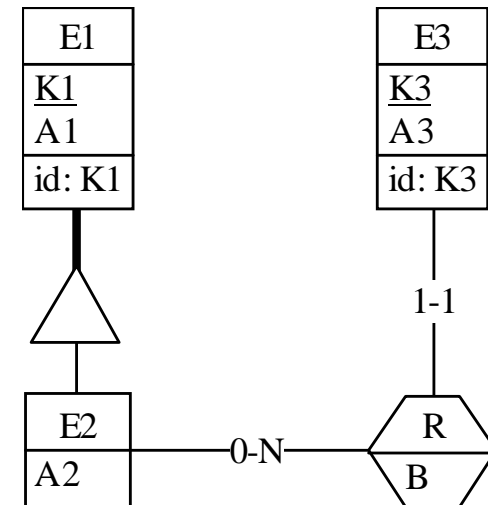
Associazioni che referenziano entità non più esistenti:

- Si consideri il seguente schema:

che viene tradotto inglobando E2 in E1
e R in E3:

```
Create table E1 (  
  K1 int not null primary key,  
  A1 int not null,  
  sel smallint not null  
    check (sel in (1,2)), -- 2 = appartiene a E2  
  A2 int,  
  constraint E2_instance  
    check ( (sel = 1 and A2 is null) or  
            (sel = 2 and A2 is not null)) );
```

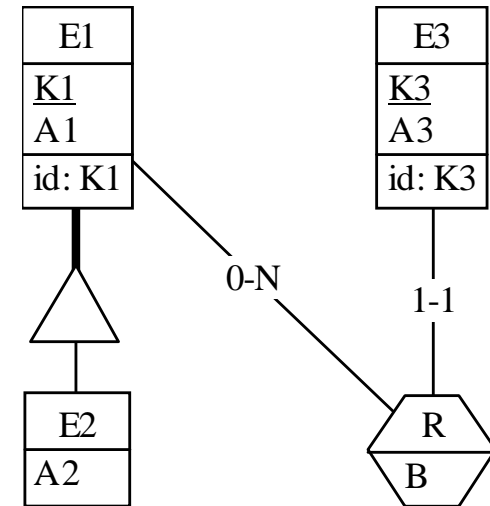
```
Create table E3 (  
  K3 int not null primary key,  
  A3 int not null  
  K1 int not null references E1,  
  B int not null);
```



Progettazione logica (2.10)

- Lo schema relazionale è impreciso, in quanto corrispondente allo schema E/R:
- Occorre quindi aggiungere un vincolo opportuno che, se si potesse disporre di clausole CHECK espressive, sarebbe:

```
Create table E3 (  
  K3 int not null primary key,  
  A3 int not null  
  K1 int not null references E1,  
  B int not null,  
  constraint R_references_E2  
    check (exists (select * from E1 where E1.K1 = K1 and sel = 2)) );
```



- Si deve fare quindi uso della query di verifica, da eseguire prima di ogni inserimento in E3:

```
-- OK se restituisce una tupla (exists)  
Select * from E1 where E1.K1 = :K1 and sel = 2;
```


Indici B⁺-tree (1.1)

Costruzione di un B⁺-tree:

- Si considera un B⁺-tree primario di ordine $O=1$ e si dettagliano i passi necessari ad inserire la sequenza di valori **12,23,17,9,18,10**
- Inizialmente, quando l'indice viene creato, viene allocato un nodo radice (che è anche foglia). I primi 2 valori vengono direttamente inseriti in tale nodo:

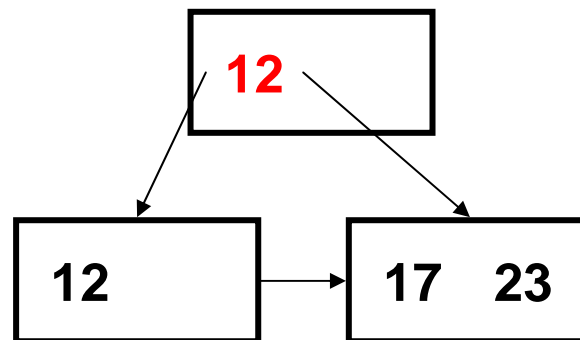
12 23

- 17** Viene allocato un nuovo nodo e collegato a quello già esistente. Le 3 ($2*O+1$) chiavi vengono ripartite ordinatamente sui 2 nodi, per convenzione inserendone 1 (O) nel nodo a sinistra e 2 ($O+1$) nel nodo a destra:

12

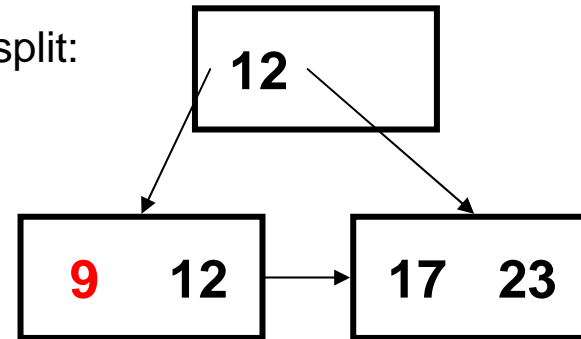
17 23

E' ora necessario creare una nuova radice, che referenzierà i due nodi foglia e conterrà come valore separatore il maggiore della foglia di sinistra (12):

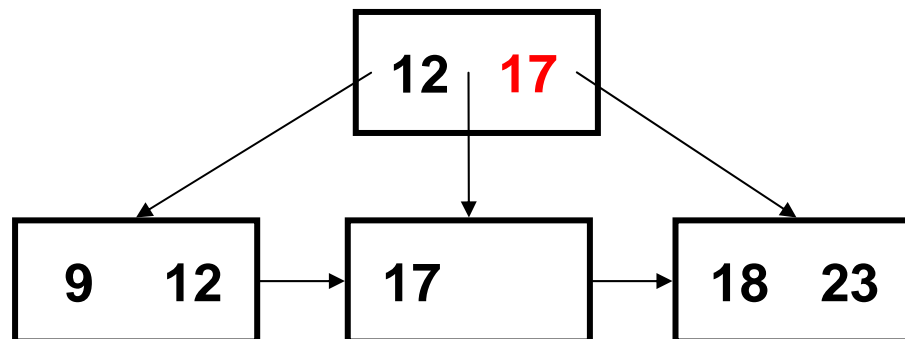


Indici B⁺-tree (1.2)

9 Viene inserito senza necessità di eseguire split:

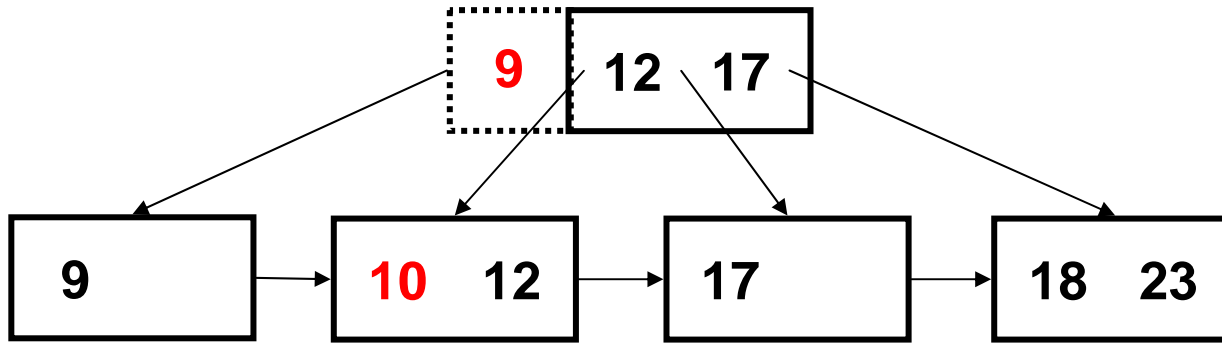


18 L'inserimento provoca lo split della seconda foglia. Il nuovo separatore (17) può essere inserito nella radice:



Indici B⁺-tree (1.3)

10 L'inserimento provoca lo split della prima foglia. Il nuovo separatore (9) **non** può essere inserito nella radice:



Il nodo radice viene quindi splittato, mantenendo gli O separatori più bassi a sinistra e gli O più alti a destra. Il separatore mediano (12) viene inserito nel nuovo nodo radice:

