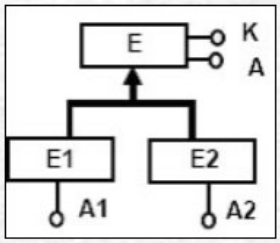
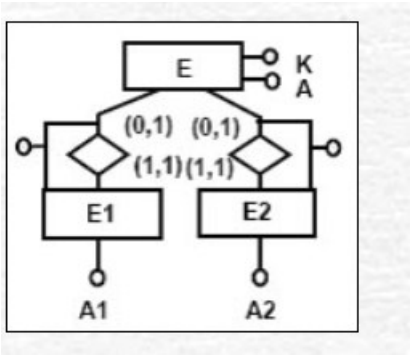
Riferimento di questo pezzo iniziale: http://www.dacrema.com/Informatica/gerarchie_elimin.htm

Per la generalizzazione esistono tre modi di eliminare le gerarchie:

Prendiamo come esempio il seguente schema E-R:

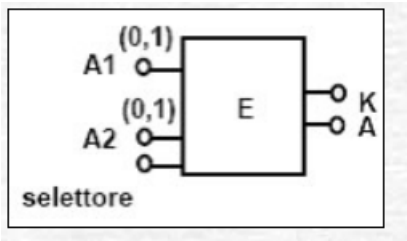


1) Mantenimento delle entità con associazioni



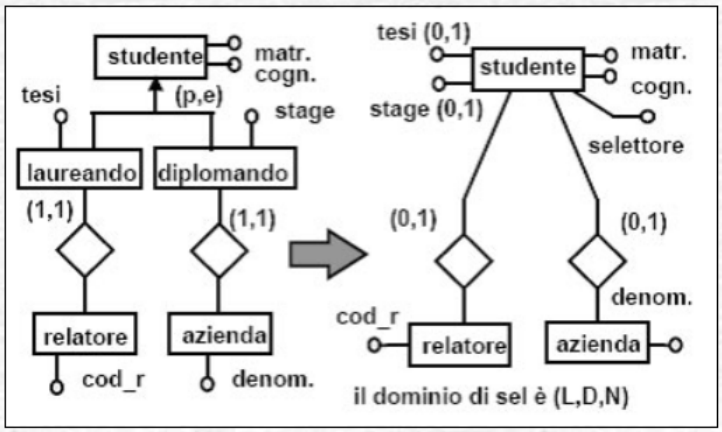
Tutte le entità vengono mantenute. Le entità figlie vengono messe in associazione con l'entità padre e sono identificate esternamente tramite l'associazione. La cardinalità (0,1) indica che per tale associazione l' entità padre può avere zero o una entità figlio. Mentre (1,1) che l' entità figlio può avere un solo padre.

2) Collasso verso l'alto



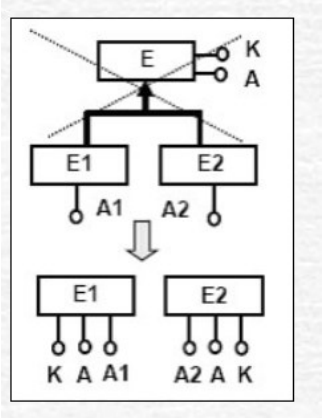
Esso riunisce tutte le entità figlie nell'entità padre.
E1 ed E2 vengono eliminate e le loro proprietà vengono aggiunte all' entità padre . Gli attributi obbligatori per le entità figlie divengono opzionali per il padre , indicato dalla cardinalità (0,1), con la conseguenza che si *avrà una certa percentuale di valori nulli*. All' entità ottenuta viene aggiunto un ulteriore attributo (selettore) che specifica se una istanza di E appartiene a una delle sottoentità.

Se la gerarchia era totale ed esclusiva il selettore ha N valori, quante sono le sottoentità. Se la gerarchia era parziale esclusiva il selettore ha N+1 valori; il valore in più serve per le istanze che non appartengono ad alcuna sottoentità. Se avessimo un overlapping occorrerebbero tanti selettori booleani quante sono le sottoentità. Vediamo di seguito un esempio di uno schema in cui è stata collassata la gerarchia verso l' alto:

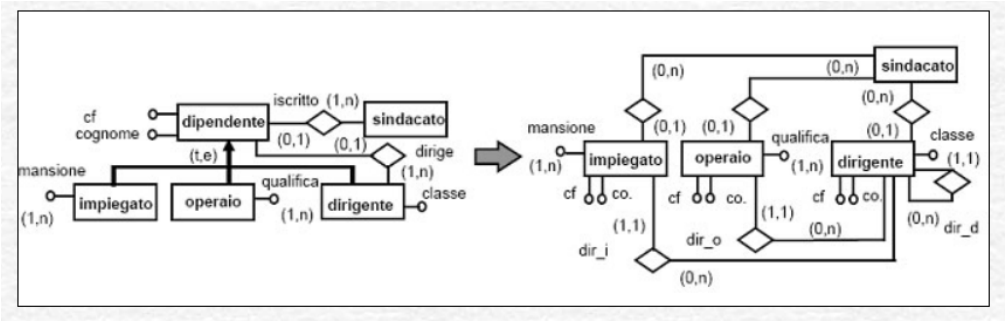


3) Collasso verso il basso

Si elimina l'entità padre trasferendone gli attributi su tutte le entità figlie. Una associazione del padre è replicata, tante volte quante sono le entità figlie *la soluzione è interessante in presenza di molti attributi di specializzazione* (con il collasso verso l'alto si avrebbe un eccesso di valori nulli).

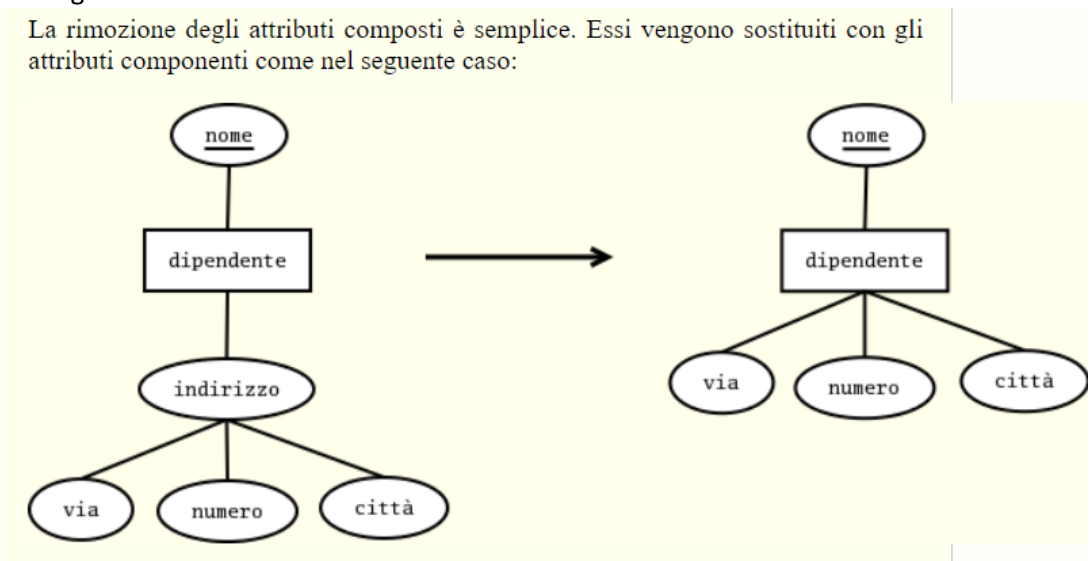


Si noti che se la copertura è parziale non si può fare il collasso verso il basso. Dove mettere gli E che non sono né E1, né E2 ? Mentre se la copertura è overlapping introduce ridondanza, infatti per una istanza presente sia in E1 che in E2 si rappresentano due volte gli attributi di E.
Vediamo di seguito un esempio di uno schema in cui è stata collassata la gerarchia verso il basso:



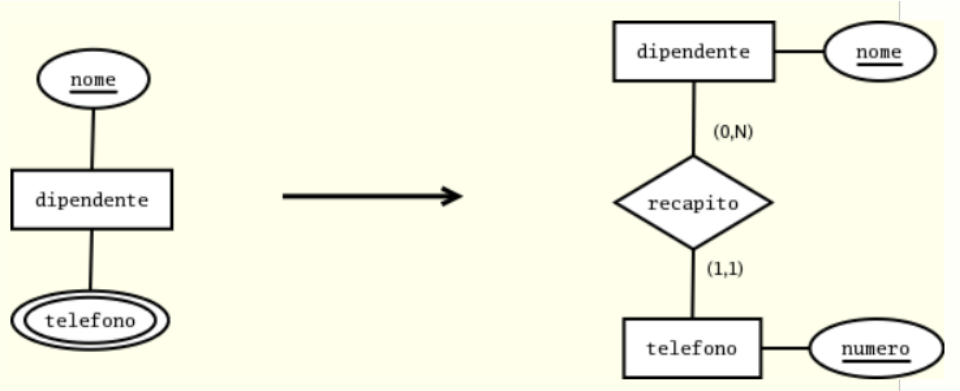
4) Nota di contorno: gli attributi multivalore e attributi composti.

Esistono campi come *Telefono* oppure *Indirizzo* per i quali sarebbe ridondante memorizzare tutti i campi in una sola tabella.
La rimozione degli attributi composti è semplice. Essi vengono sostituiti con gli attributi componenti come nel seguente caso:



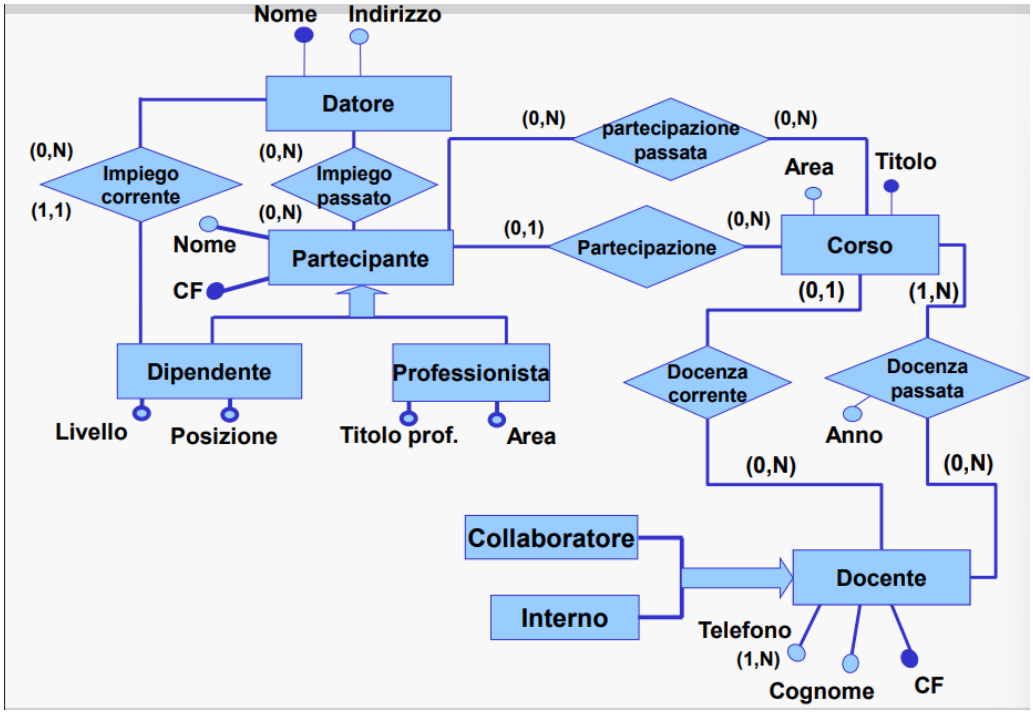
Eventuali gerarchie di attributi composti si traducono similmente procedendo dalle foglie verso la radice. A questo punto tutti gli attributi sono semplici, derivati, oppure multivalore.

Per quanto riguarda invece il campo Telefono che è multivalore, si crea una nuova entità che contiene i valori dell'attributo e la si collega all'entità che possedeva l'attributo mediante una nuova relazione uno a molti o molti a molti, a seconda dei casi.



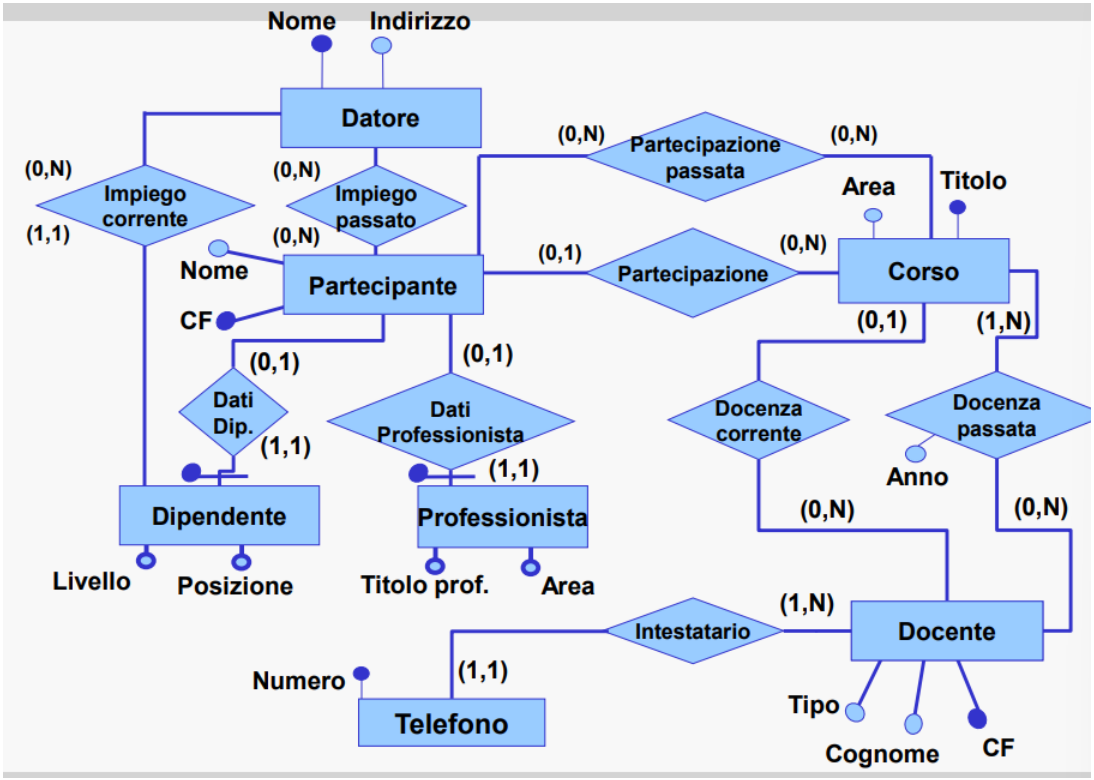
Se l'attributo telefono è opzionale, il vincolo di cardinalità (1,1) va sostituito con (0,1). Se inoltre un numero di telefono può essere condiviso da più persone (ad esempio, il telefono di casa), allora il vincolo di cardinalità (1,1) va sostituito con (0,N).

L'esempio visto in classe parte dallo schema:



E attua una minimizzazione dei valori nulli:

- mantenendo Docenza Corrente e Docenza Passata come relazioni
- usando il mantenimento delle entità con associazioni tra Partecipante e Dipendente/Professionista
- creando per l'attributo multivalore Telefono, un'entità apposita
- mantenendo Partecipazione e Partecipazione Passata come relazioni e similmente anche per Impiego (Corrente e Passato)



Creando come schema:

- Partecipante(CF, Nome, Corso-Attuale)
    - Partecipante.Corso-Attuale → Corso.Titolo
- Datore(Nome, Indirizzo)
- Corso(Titolo, Area, CF-Docente-Corrente)
    - Corso.CF-Docente-Corrente → Docente.CF
- Docente(CF, Cognome,Tipo)
    - Docente.Tipo ∈ { Collaboratore, Interno }
- Telefono(Numero, CF-Docente)
    - Telefono.CF-Docente → Docente.CF
- Dipendente(CF, Livello, Posizione, Datore)
    - Dipendente.CF → Partecipante.CF
    - Dipendente.Datore → Datore.Nome
- Professionista(CF,Titolo-Prof, Area)
    - Professionista.CF → Partecipante.CF
- ImpiegoPassato(CF-Partecipante,Nome-Datore)
    - ImpiegoPassato.CF-Partecipante → Partecipante.CF
    - ImpiegoPassato.Nome-Datore → Datore.Nome
- PartecipazionePassata(CF-Partecipante,Titolo-Corso)
    - PartecipazionePassata.CF-Partecipante → Partecipante.CF
    - PartecipazionePassata.Titolo-Corso → Corso.Titolo
- DocenzaPassata(CF-Docente,Titolo)
    - DocenzaPassata.CF-Docente → CF.Docente
    - DocenzaPassata.Titolo → Corso.Titolo

Esercizi concreti

Il sistema è condiviso tra diverse istituzioni, di ognuna delle quali è di interesse sapere il nome dell'istituzione, la via e il codice ISTAT del comune in cui ha la sede principale.

Per ogni istituzione, si vuole sapere le persone che lavorano presto tale istituzione ed i bandi di concorso che ha eventualmente emesso.

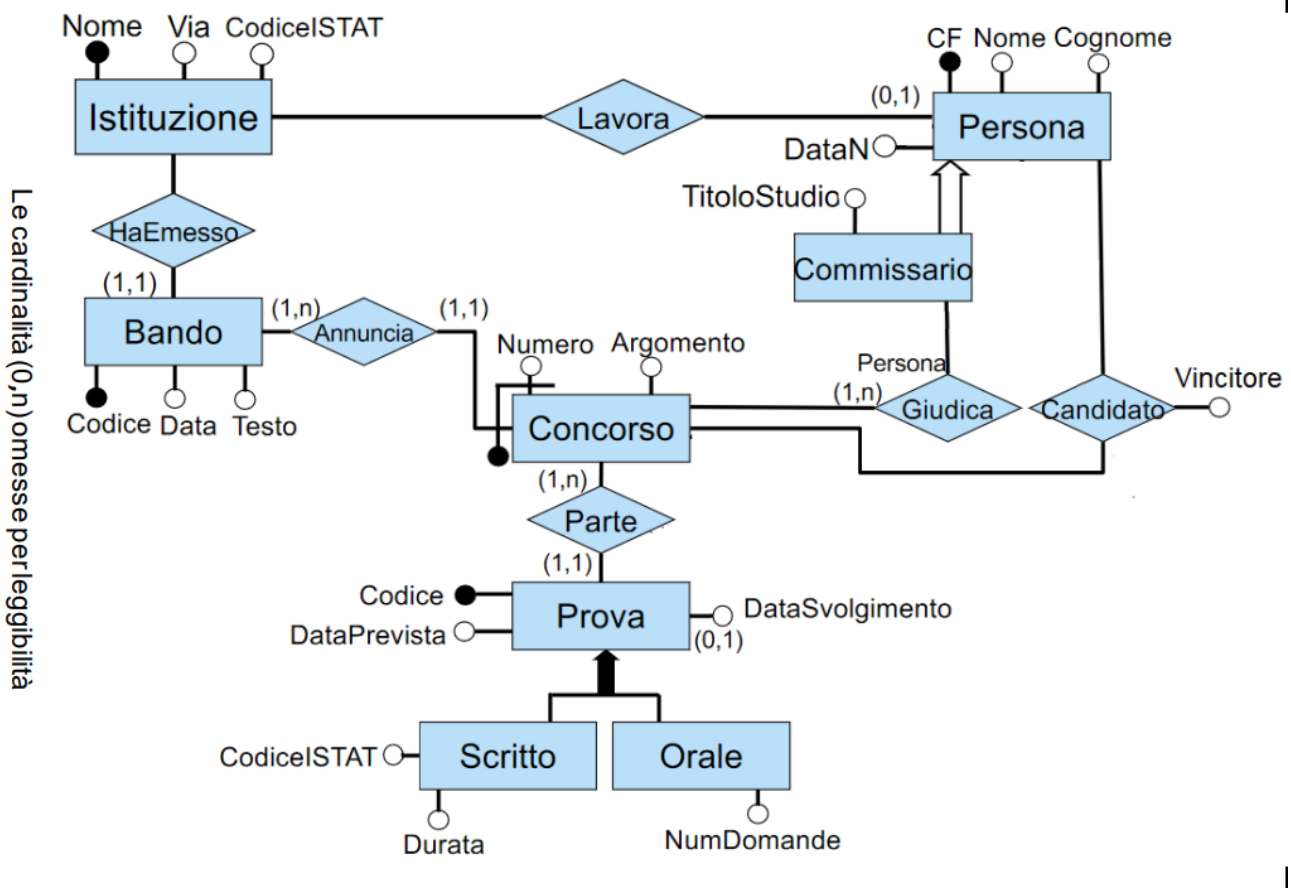
Di ogni persona interessa il codice fiscale (identificativo), il nome, il cognome, la data di nascita. Una persona può lavorare presso una istituzione (ma anche essere disoccupata). Alcune persone possono anche fungere da commissari giudicatori di concorso: per loro, si vuole sapere quale sia il titolo di studio.

Di ogni bando interessa il codice identificativo, la data di pubblicazione, il testo ed i concorsi (almeno uno) che sono annunciati nel bando stesso.

Di ogni concorso interessa il bando in cui è annunciato, il numero (unico nell'ambito del bando in cui è annunciato ma diversi bandi possono avere lo stesso numero concorso), l'argomento del concorso (giurisprudenza, ingegneria civile, amministrazione, ecc.) e le prove di cui è composto (almeno una), indentificata da un codice identificativo. Per ogni prova, si vuole conosce la data di svolgimento prevista e successivamente la data di svolgimento effettiva, quando questa avviene.

Le prove sono o scritte o orali e di ogni prova scritta interessa anche la durata e il codice ISTAT del comune dove si prevede si svolga, mentre di ogni prova orale interessa anche il numero di domande previste.

Di ogni concorso interessano anche le persone (almeno una) che hanno partecipato come commissari, le persone che hanno partecipato come candidati (se ci sono) e, tra queste ultime, quelle che sono risultate vincitrici (se ci sono).



Occorre ottimizzare la seguente operazione: quando si accede alla durata di una prova o al numero di domande, si vuole sempre conoscere il concorso di cui fa parte.

Quindi si opera eliminando l'entità *Prova* e duplicando i suoi attributi per *Scritto* ed *Orale*.
In merito a Commissario, si mantiene una relazione con cardinalità (0,1) tra Persona e Commissario (perché una persona può essere Commissario) e (1,1) tra Commissario e Persona (perché un commissario è sempre una persona).
Candidato rimane una relazione, essendoci due cardinalità (0,n) e (0,n), aggiungendo solo Vincitore come attributo (può essere nullo in un primo momento come attributo, dato che il testo indica che *può* esserci un vincitore; similmente, essendo che *possono* esserci dei candidati, nella stessa relazione candidato, potrà essere nullo anche l'attributo Persona)
Essendo inoltre che una Persona può essere disoccupata, può essere nullo il collegamento di chiave esterna con Istituzione.
In nome di tutto ciò, si struttura così lo schema logico (indicando con * i valori nulli)

Istituzione (<u>Nome</u>, CodiceISTAT, Via)
Persona (<u>CF</u>, Nome, Cognome, Data_nascita, Istituzione*)
FK:
Persona.Istituzione → Istituzione.Nome

Commissario (<u>CF</u>, Titolo_studio, Persona)
FK:
Commissario.CF → Persona.CF

Giudice (Commissario, Concorso)
FK:
Giudice.Commissario → Commissario.CF
Giudce.Concorso → Concorso.Numero

Candidato (Persona, Vincitore*, Concorso)
FK
Cnndidato.Persona → Persona.CF
Candidato.Concorso → Concorso.Numero

Concorso (<u>Numero</u>, Argomento, Bando)
FK
Concorso.Numero → Bando.Codice

Bando (<u>Codice</u>, Data_pubb, Testo, Istituzione)
FK
Bando.Istituzione → Istituzione.Nome

Prova_scritta (<u>Codice</u>, DataPrevista, DataSvolgimento, CodiceISTAT, Durata, Concorso)
FK
Prova_scritta.Concorso → Concorso.Numero

Prova_orale (<u>Codice</u>, DataPrevista, DataSvolgimento*, NumDomande, Concorso)
FK
Prova_orale.Concorso → Concorso.Numero

Si richiede di progettare lo schema concettuale Entità-Relazione di un'applicazione relativa ai festival musicali organizzati da agenzie.

Di ogni agenzia interessa il nome (identificativo), la provincia della sede (con codice ISTAT identificativo e regione), la persona che la dirige (con anno di inizio direzione) e gli avvisi di festival che ha eventualmente pubblicato.

Festival sono annunciati tramite avvisi. Di ogni avviso interessa l'agenzia che lo ha pubblicato, il codice (unico nell'ambito dell'agenzia che lo ha pubblicato), la data di pubblicazione ed il testo definitivo.

Gli avvisi si suddividono in annullati e validi. Dei primi interessa il motivo dell'annullamento e dei secondi interessa il festival che è annunciato nell'avviso.

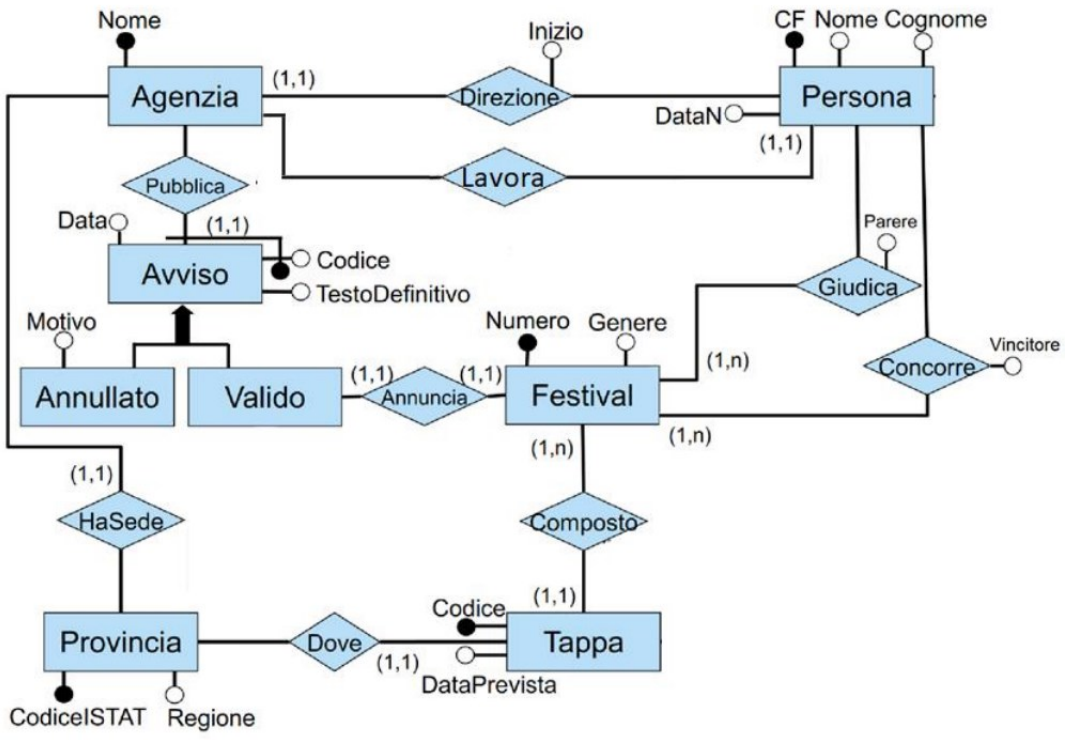
Di ogni festival interessa l'avviso in cui è annunciato, il numero (identificativo), il genere di musica coperto (ad esempio, classica, pop, rock, ecc.) e le tappe di cui è composto (almeno una), ciascuna con codice identificativo(univoco in assoluto), data di svolgimento prevista e la provincia in cui svolgerà.

Di ogni festival interessa sapere le tappe (almeno una), le persone che hanno partecipato come giurati (almeno una), le persone che hanno partecipato come concorrenti (almeno una) e, tra queste ultime, quelle che sono risultate vincitrici (se ci sono).

Di ogni persona interessa il codice fiscale (identificativo), il nome, il cognome e la data di nascita.

Alcune note:

- Provincia va messa come entità essendo attributo multivalore e, in particolare, perché indicata dal testo come contenitore di più campi (Codice Istat e Regione)
- La solita entità Persona collega i Giudici (di cui salva un parere anche se non richiesto dal testo), i concorrenti (mettendo come al solito dentro il campo vincitore), e il campo Direzione (Lavora direi che fa parte di una linea poi tagliata dal testo; quindi, salvare se per il giurato interessa anche l'agenzia di cui fa parte)



A partire dal Diagramma ER dell'Esercizio 1, produrre uno schema relazionale del database nel riquadro sottostante, minimizzando i valori nulli delle relazioni. Indicare i vincoli di chiave e gli attributi che ammettono valori nulli. Illustrare come ristrutturare l'ER per essere traducibile in uno schema relazionale.

Per minimizzare i valori nulli, pensiamo alle generalizzazioni, in questo caso su Avviso.
A tale scopo, si inserisce due relazioni, avendo cardinalità (0,1) da parte di Avviso e cardinalità (1,1) da parte di Annullato e da parte di Valido, tale che quest'ultimo sia direttamente collegato a Festival.
Togliendo Lavora perché parte di una vecchia richiesta del testo, tutto il resto (Direzione/Giudici/Concorso) vengono pensate come relazioni.

Agenzia (Nome, Dirigente, Provincia)
FK:
Agenza.Dirigente → Direzione.Persona
Agenzia.Provincia → Provincia.CodiceISTAT

Direzione (Data_inizio, Dirigente)
FK:
Direzione.Dirigente → Persona.CF

Persona (CF, Nome, Cognome, DataN)

Giudica (Giudice, Parere, Festival)
FK:
Giudica.Giudice → Persona.CF
Giudica.Festival → Festival.Numero

Concorre (Vincitore, Festival)
FK:
Concorre.Vincitore → Persona.CF
Concorre.Festival → Festival.Numero

Festival (Numero, Genere, Avviso_valido)
FK:
Festival.Avviso_valido → Avviso_valido.Avviso

Avviso (Codice, TestoDefinitivo, Data, Agenzia)
FK:
Avviso.Agenzia → Agenzia.Nome

Avviso_annullato (Avviso, Motivo)
FK:
Avviso_annullato.Avviso → Avviso.Codice

Avviso_valido (Avviso, Festival)
FK:
Avviso_valido.Avviso → Avviso.Codice
Avviso_valido.Festival → Festival.Numero

Provincia (CodiceISTAT, Regione)

Tappa (Codice, Data_prevista, Festival)
FK:
Tappa.Festival → Festival.Numero

Si richiede di produrre lo schema concettuale Entità-Relazione di un'applicazione relativa alla gestione delle edicole per la vendita di giornali, Disegnare il diagramma ER nel riquadro della pagina che segue.

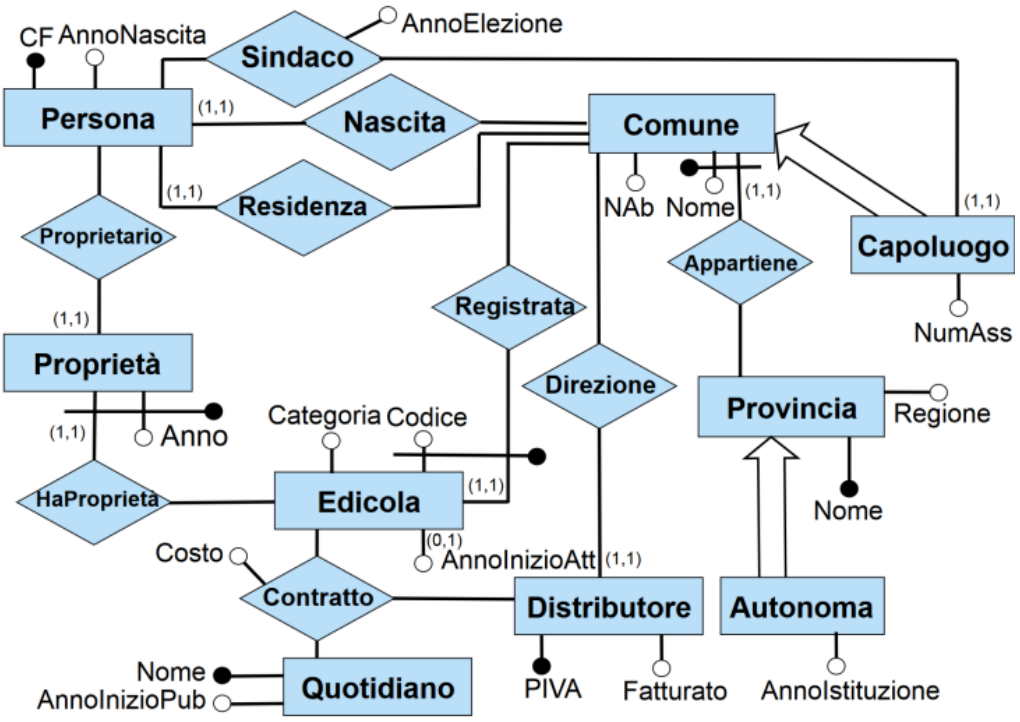
Si richiede di progettare lo schema ER concettuale di un'applicazione relativa alle edicole per la vendita di giornali.
Di ogni edicola interessa il comune in cui essa è registrata, il codice, che è unico nell'ambito del comune in cui l'edicola stessa è registrata, la categoria, l'anno di inizio attività (non sempre disponibile), e i contratti che l'edicola ha con i distributori per l'approvvigionamento dei quotidiani.

Ogni contratto riguarda un'edicola, un quotidiano ed un distributore, ed è caratterizzato dal costo mensile a carico dell'edicola. Infine, per ogni edicola, interessa conoscere le varie persone che sono state proprietarie dell'edicola nei diversi anni, tenendo conto del fatto che in ogni anno un'edicola ha al massimo un proprietario.

Di ogni persona interessa il codice fiscale (id), l'anno di nascita, il comune di nascita, ed il comune di residenza. Di ogni distributore di quotidiani interessa la partita IVA (id), il fatturato ed il comune in cui è situata la direzione.

Di ogni quotidiano interessa il nome (identificativo), e l'anno di inizio pubblicazione. Di ogni comune interessa la provincia di appartenenza, il nome (unico nella provincia), ed il numero di abitanti. Dei comuni che sono capoluogo di provincia interessa l'attuale sindaco (con l'anno di elezione), ed il numero di assessori comunali.

Di ogni provincia interessa il nome (identificativo) e la regione di appartenenza. Alcune province sono "autonome" e di esse interessa anche l'anno di istituzione



A partire dallo ER concettuale dell'Esercizio 1, produrre uno schema relazionale del database nel riquadro sottostante, minimizzando i valori nulli. Indicare la chiave primaria, i vincoli di chiave esterne e gli attributi che ammettono valori nulli. Illustrare come ristrutturare il diagramma ER per essere direttamente traducibile in uno schema relazionale.[1]

Soluzione (Parziale) Esercizio 2
Circa l'esercizio 1. Non è possibile rappresentare la proprietà come una relazione, poiché una persona può essere proprietaria di una edicola per anni successivi, mentre una relazione permette solamente una tupla (p,e,a) con lo stesso valore di persona *p* e di edicola *e*. D'altronde, una cardinalità (1,N) da entrambi i lati non funzionerebbe neanche, perché permetterebbe a più persone di essere propretarie di una edicola nello stesso anno. **Ristrutturazione.** Volendo minizzare i valori nulli, occorre sostituire le due generalizzazioni con due relazioni con cardinalità (1,1) dal lato di Capolugo e Autonomia e (0,1) dal lato di Comune e Provincia. Gli identificatori di Comune e Autonomia sono le relazioni stesse appena introdotte.

Il sistema deve memorizzare le informazioni di ogni pista della catena. Di ogni pista, è di interesse sapere (1) la città in cui si trova, (2) l'anno di apertura e (3) il fatturato. Ogni città non può avere più di una pista. Inoltre, ogni pista è gestita da una persona, però una pista potrebbe temporaneamente non avere una persona che la gestisce. Si noti che una persona non può prendere in gestione più di una pista della catena.

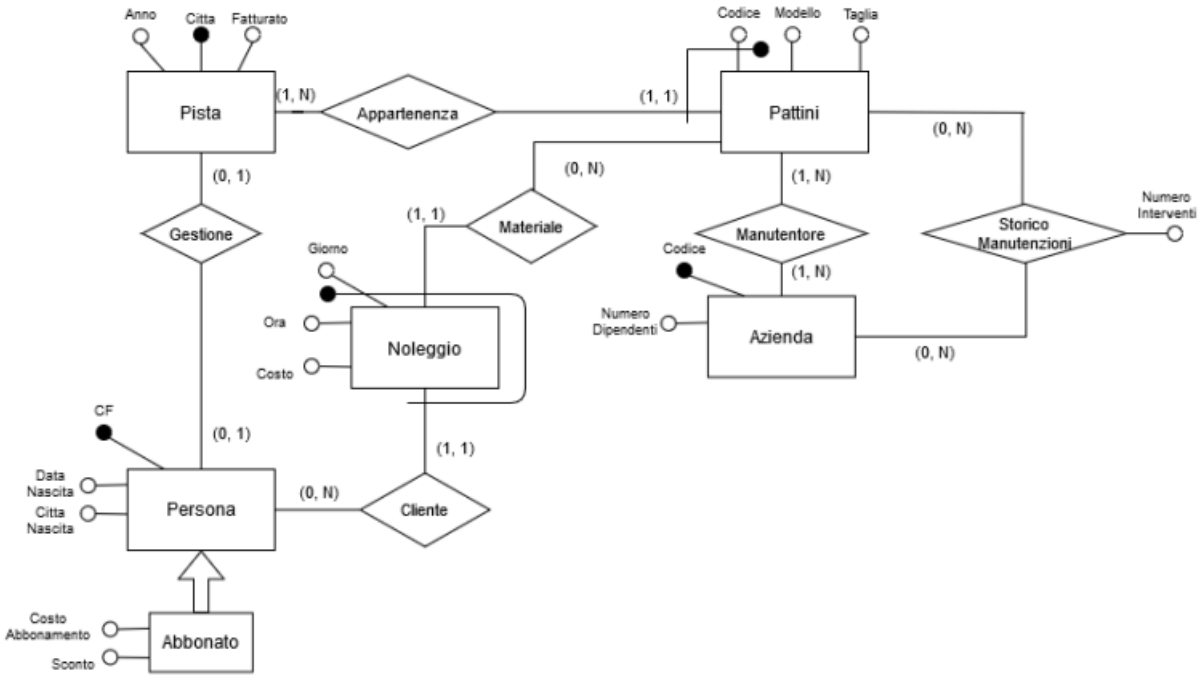
Il sistema memorizza un'anagrafe delle persone: di ogni persona, è di interesse sapere il codice fiscale (identificativo), la data di nascita e la città di nascita.

Quando le persone desiderano essere clienti di una pista, noleggiano un paio di pattini. Ogni paio di pattini è associato ad esattamente una pista.

Di ogni paio di pattini interessa il codice (unico all'interno della pista), il modello, la taglia. Inoltre, per ogni paio di pattini, è di interesse sapere quali aziende sono in grado di farne manutenzione. Si vuole anche memorizzare lo storico delle manutenzioni: per ogni paio di pattini, è di interesse sapere quante volte (intero) il paio è andato in manutenzione in ogni azienda in grado di manutenere. Di ogni azienda di riparazione pattini interessa il codice identificativo e il numero di dipendenti.

Come detto, ogni paio di pattini è noleggiato da persone per usarle nella pista associata. In particolare, si vuole sapere per ogni noleggio: (1) quale paio è stato preso, (2) il giorno e l'ora del noleggio e (3) il costo complessivo del noleggio. Si noti che ogni paio di pattini può essere noleggiato più volte dalla stessa persona, ma solo in giorni diversi.

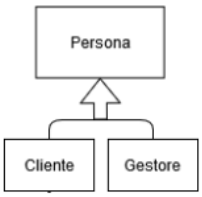
Alcune persone sono clienti abbonati. Per tali clienti, è di interesse sapere il costo dell'abbonamento annuale (unico per cliente) e quale sconto (in percentuale) ottengono nei noleggi in virtù del loro abbonamento.



Si noti che:

- si parla di aziende in grado di fare manutenzione; ciò presuppone che ci sia almeno una azienda che fa manutenzione e, nel corso del tempo, avrò diverse aziende che fanno manutenzione; quindi è necessaria la doppia relazione
- la possibilità di scrivere Cliente e Gestore come generalizzazione (totale/parziale) di Persona ha poco senso; si noti che, a prescindere, volendo minimizzare i valori nulli si dovrebbero creare delle relazioni e i clienti non hanno alcun dato che non sia già presente in Persona.
  Abbonato avrebbe dei dati, ma appunto la ristrutturazione considera l'introduzione di una relazione con cardinalità (0,1) ed (1,1) tra Persona ed Abbonato.
  Infatti, il prof dice questo:

L'ultima osservazione riguarda la possibilità qui a fianco (con *Persona* che generalizza le entità *Cliente* e *Gestore*). Tale soluzione è errata a meno che viene esplicitamente detto che la generalizzazione persona-cliente-gestore è sovrapposta. Se non viene detto che è *sovrapposta* viene considerata il default, che è *esclusiva*: vorrebbe dire che una persona può essere solamente cliente oppure gestore. Si noti che il fatto che la generalizzazione sia totale (freccia piena) o parziale (freccia vuota) non cambia nulla.



Considerato inoltre che una persona gestisce una sola pista ed una pista ha un solo gestore, allora questa sarà normalmente collegata come chiave esterna sotto forma di relazione.

Lo schema logico è:
PERSONA(<u>CF</u>, DataNascita, CittaNascita)
PISTA(<u>Citta</u>, Fatturato, Anno, Gestore)

- Chiave Esterna: PISTA.Gestore → Persona.CF
- Chiave Aggiuntiva: Gestore

PATTINI(<u>Codice</u>, <u>CittaPista</u>, Modello, Taglia)

- Chiave Esterna: PATTINI.CittaPista → PISTA.Citta

NOLEGGIO(<u>CodicePattini, CittaPista, Persona, Giorno</u>, Ora, Costo)

- Chiave Esterna: NOLEGGIO.(CodicePattini, CittaPista) → PATTINI.(Codice,CittaPista)
- Chiave Esterna: NOLEGGIO.Persona → Persona.CF A

AZIENDA(<u>Codice</u>, NumeroDipendenti)

MANUTENTORE(<u>CodicePattini, CittaPista, CodiceAzienda</u>)

- Chiave Esterna: MANUTENTORE.(CodicePattini, CittaPista) → PATTINI.(Codice,CittaPista)
- Chiave Esterna: MANUTENTORE.CodiceAzienda → AZIENDA.Codice

STORICO(<u>CodicePattini, CittaPista, CodiceAzienda</u>, NumeroInterventi)

- Chiave Esterna: STORICO.(CodicePattini, CittaPista) → PATTINI.(Codice,CittaPista)
- Chiave Esterna: STORICO.CodiceAzienda → AZIENDA.Codice

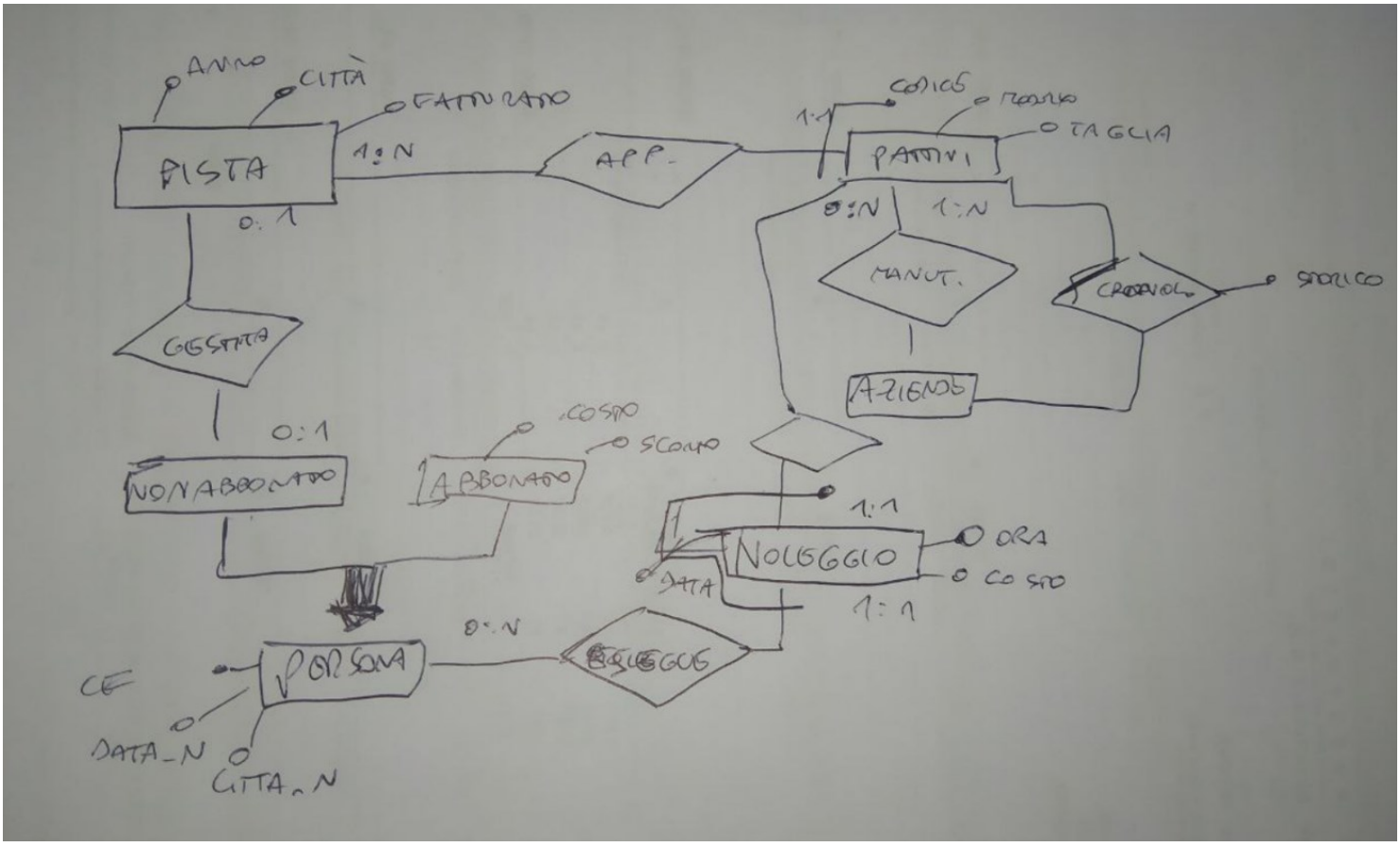
ABBONATO(<u>CF</u>, Costo, Sconto)

- Chiave Esterna: ABBONATO.CF → PERSONA.CF

Un'altra possibile soluzione discute questo fatto:

- con questa soluzione si ammette che un abbonato possa gestire una pista; il testo non cita esplicitamente questo fatto. Per questo, la persona ammette che sia abbonato che non abbonato possano gestire. Un cliente concettualmente non aggiunge alcuna informazione utile che non sia già in persona; rimane relazione

Più corretto quindi eseguire una generalizzazione totale tra Persone e Abbonato e Non Abbonato, ammettendo che un Non Abbonato gestisca e che il Gestore sia una persona non abbonata e che ha facoltà di gestire.



Il negozio di fumetti vuole costruire un database relativo ai fumetti in vendita nel negozio, tenendo traccia dei fumetti venduti e dei clienti che hanno effettuato acquisti.

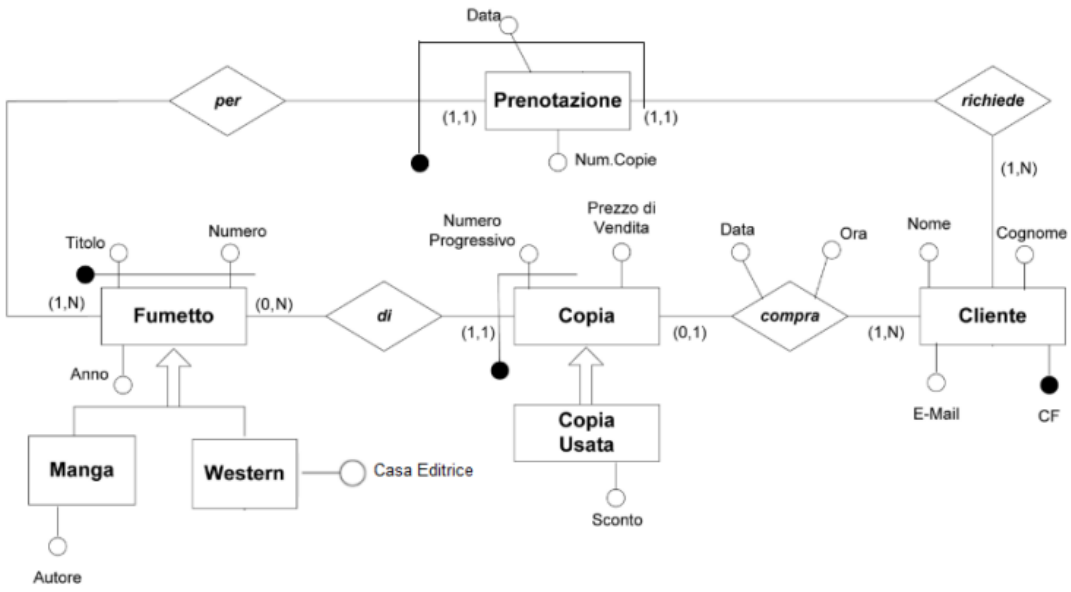
Ogni fumetto è identificato dal nome della serie di cui fa parte e dal numero che lo contraddistingue all'interno della serie (ad es. Topolino n.21243). Per tutti i fumetti si vuole inoltre conoscere l'anno di pubblicazione.

Per alcune tipologie di fumetti sono di particolare interesse informazioni addizionali. Per i manga (i fumetti giapponesi) interessa conoscere l'autore, mentre per i fumetti western è interesse conoscere la casa editrice.

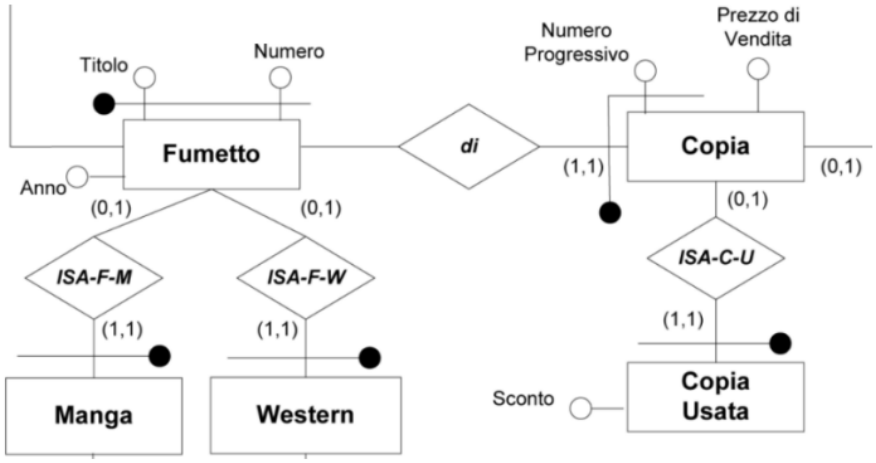
Il negozio vuole tener traccia delle varie copie disponibili per ciascun fumetto; ciascuna copia è identificata un numero progressivo e da un prezzo diverse. Copie diverse dello stesso fumetto possono avere prezzi di vendita diverse (per esempio, in funzione del momento di stampa della copia).

Ciascun cliente (di cui interessa conoscere il codice fiscale, il nome, il cognome e l'e-mail) può fare le seguenti cose:
- Acquistare copie di fumetti: in tal caso dell'acquisto si memorizza la data e l'ora dello stesso
- Prenotare fumetti di interesse: un cliente può prenotare un certo numero di copie di un particolare fumetto, memorizzando data della prenotazione. Lo stesso cliente può prenotare lo stesso fumetto più volte, ma solo in date diverse.

Si noti che il negozio può vendere anche copie usate di un fumetto, e in questo caso è di interesse conoscere lo sconto applicato rispetto alla versione venduta a prezzo pieno.



La ristrutturazione dello schema logico le vede come al solito come delle "is-a" e come tale aggiunge relazioni di collegamento intermedie:



E lo schema logico completo è il seguente:

FUMETTO (Titolo, Numero, Anno)

MANGA (Titolo, Numero, Autore)

- Chiave Esterna: MANGA.(Titolo,Numero) → FUMETTO.(Titolo,Numero)

WESTERN (Titolo, Numero, CaseEditrice)

- Chiave Esterna: WESTERN.(Titolo,Numero) → FUMETTO.(Titolo,Numero)

CLIENTE(CF, Nome, Cognome, E-mail)

COPIA(FumettoTitolo, FumettoNumero, NumeroProgressivo, PrezzoVendita)

- Chiave Esterna: COPIA.(FumettoTitolo, FumettoNumero) → FUMETTO.(Titolo,Numero)

COPIA-USATA(FumettoTitolo, FumettoNumero, NumeroProgressivo, Sconto)

- Chiave Esterna: COPIA-USATA.(FumettoTitolo, FumettoNumero,NumeroProgressivo) → COPIA.(FumettoTitolo, FumettoNumero,NumeroProgressivo)

COMPRA(FumettoTitolo, FumettoNumero, NumeroProgressivo, CF-Cliente, Data, Ora)

- Chiave Esterna: COMPRA.(FumettoTitolo, FumettoNumero,NumeroProgressivo) → COPIA.(FumettoTitolo, FumettoNumero,NumeroProgressivo)
- Chiave Esterna: COPIA.CF_Cliente → CLIENTE.CF

PRENOTAZIONE(FumettoTitolo, FumettoNumero, CF-Cliente, NumCopie)

- Chiave Esterna: PRENOTAZIONE.(FumettoTitolo, FumettoNumero) → FUMETTO.(Titolo,Numero)
- Chiave Esterna: PRENOTAZIONE.CF-Cliente → CLIENTE.CF

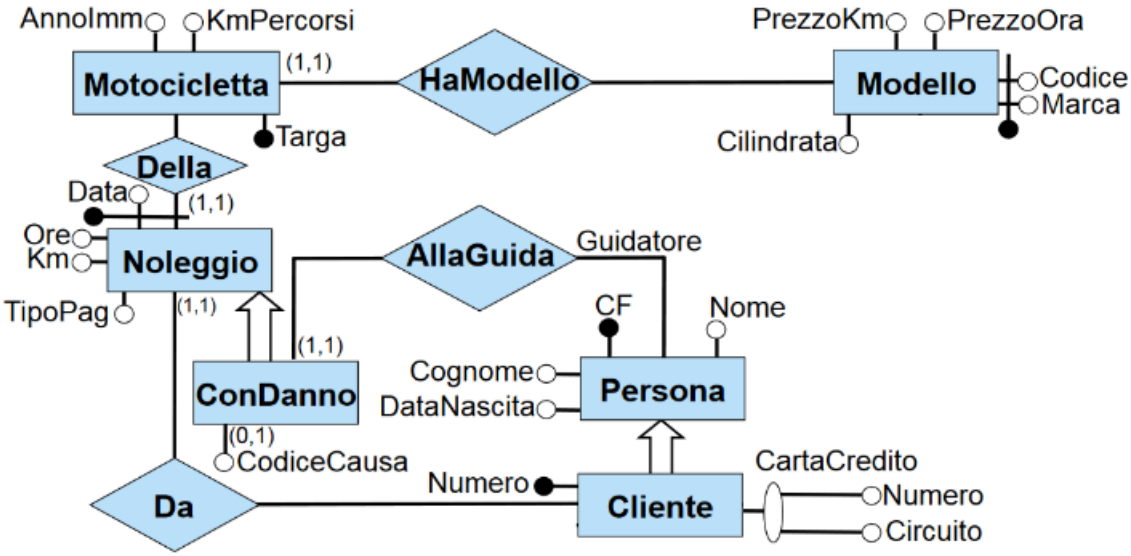
Si noti che, se COMPRA fosse fusa con COPIA, il numero di attributi con valori potenzialmente nulli sarebbero alti. Infatti, COPIA avrebbe tre attributi CF-Cliente, Data e Ora che avrebbero valori nulli per tutte le copie non ancora vendute.

Il negozio possiede tante motociclette. Di ogni motocicletta, interessa la targa (identificativo), il modello, l'anno di immatricolazione ed i chilometri percorsi. Di ogni modello interessala marca (ad esempio, "Honda"), il codice (unico nell'ambito della marca), la cilindrata, il prezzo del noleggio all'ora ed il prezzo del noleggio al chilometro.

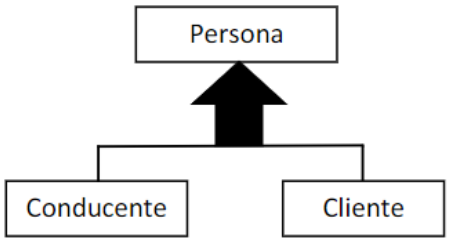
Di ogni noleggio, interessa il cliente che l'ha effettuato, la motocicletta noleggiata, la data in cui è avvenuto (ogni noleggio è infatti relativo ad un giorno), la durata in ore, la tipologia di pagamento (a chilometro oppure ad ora) ed i chilometri percorsi. Si noti che una stessa motocicletta non può essere noleggiata più volte nello stesso giorno.

Dei clienti, Il negozio vuole conoscere il numero di telefono. Ogni cliente possiede una carta di credito, di cui interessa il numero ed il circuito.

Per quei noleggi per i quali si è verificato un danno alla motocicletta noleggiata, interessa il codice indicante la causa del danno (se nota) e la persona-conducente che era alla guida della motocicletta al momento del verificarsi del danno. Di ogni conducente, l'applicazione deve memorizzare i seguenti dati: codice fiscale (identificativo), nome, cognome e data di nascita. Si noti che il conducente può coincidere con il cliente del noleggio oppure essere un'altra persona.



Si noti che è errata la seguente soluzione in cui gli attributi di Persona rimangono quelli attuali ed, inoltre, *Conducente* è in relazione *AllaGuida* con *ConDanno* e *Cliente* è in relazione con *Noleggio*:



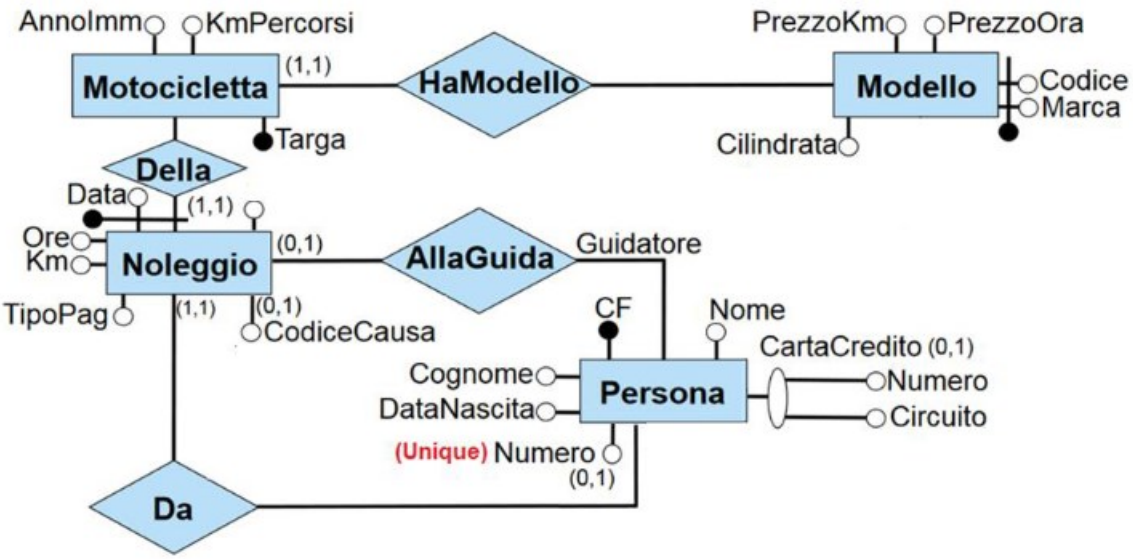
Infatti, questo implicherebbe che un conducente non può essere cliente (si pensi al caso uomo-donna). La generalizzazione parziale o totale (freccia vuota o piena) è irrilevante rispetto all'esclusività di essere conducente o cliente.

Ristrutturazione:

Dato che il testo indica questo:
Nella progettazione logica, si vuole assicurare il minimo numero minimo di tabelle necessarie, anche a costo di valori nulli. Questo vincolo è al fine di minimizzare il numero di join.

Siamo in presenza di una ristrutturazione semplice:
- Tutti gli attributi di Con Danno entrano a far parte di Noleggio
- Tutti gli attributi di Cliente entrano a far parte di persona

SI porta quindi tutto su:



Il vincolo *Unique* per numero serve a rappresentare che, se una persona ha un numero telefonico, allora è l'unica con quel numero. Questo viene derivato dal fatto che l'entità Cliente nel diagramma ER concettuale usa tale attributo come identificatore. Si noti che una persona potrebbe non avere un numero di telefono perché solo le persone clienti lo hanno

Da Esercitazione sull'ER:

La serra *PianteBellissime* vuole dotarsi di un sistema informativo per gestire la manutenzione e la vendita delle piante. La serra è specializzata in piante di alto fusto, ma vende anche piante fiorite.

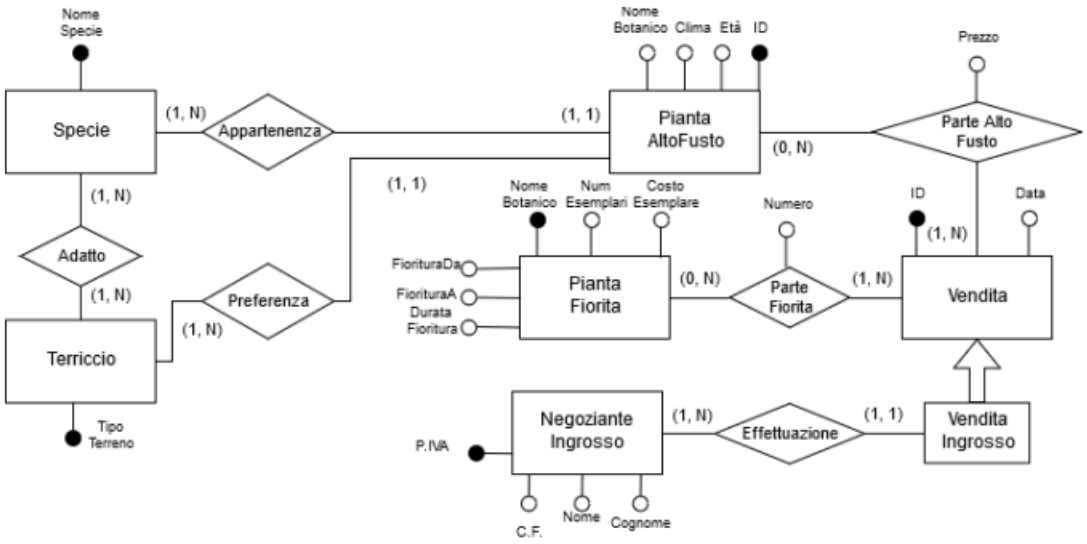
Per le piante ad alto fusto, si vogliono memorizzare i singoli esemplari in magazzino. Ogni esemplare è contraddistinto da un codice individuale, la specie, il nome botanico, l'età in anni, il clima ed il tipo di terriccio ideali.

Per le piante fiorite, non è di intesse memorizzare i singoli esemplari, ma solo le quantità a disposizione in forma aggregata. Di conseguenza, I tipi di piante fiorite non hanno un codice individuale, sono raggruppate in insiemi identificati tramite il nome botanico; di ogni insieme sono noti il periodo e la durata della fioritura, il numero di esemplari in magazzino, nonché il costo del singolo esemplare (uguale per tutti gli esemplari).

Le piante di alto fusto, se non vendute, devono essere sottoposte ad un trattamento annuale di svasamento con terricci diversi a seconda del tipo. A tale scopo, occorre associare ogni specie di pianta ad uno o più terricci.

La serra ha una clientela di due tipi: singoli individui che acquistano al dettaglio e negozianti che acquistano all'ingrosso. Soltanto per questi ultimi è necessario conoscere nome, cognome, codice fiscale e partita iva.

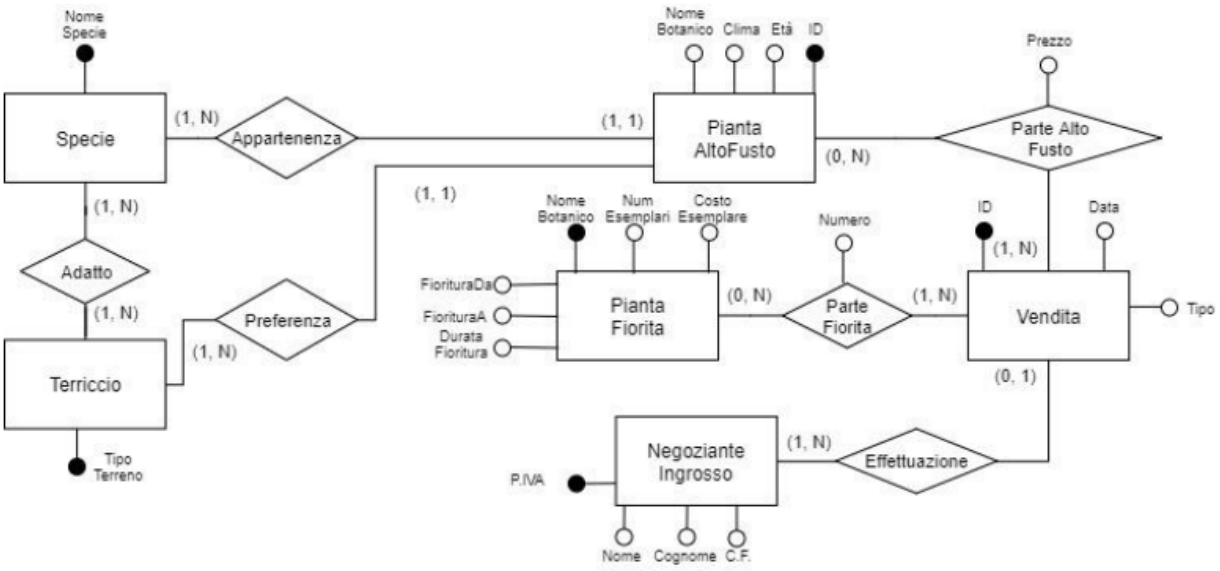
Di ogni vendita interessa memorizzare la data, l'identificativo, le piante acquistate. Ogni vendita contiene uno o più tipi di piante fiorite e piante ad alto fusto. Per ogni vendita, ogni tipo di pianta fiorita è venduto in una certa quantità; viceversa, ogni vendita contiene solo un esemplare di un tipo di pianta ad alto fusto, di cui si vuole conoscere il prezzo di vendita, che può cambiare ad ogni vendita. Solo per le vendite all'ingrosso, si interessa conoscere l'acquirente.



In merito all'ER alcune osservazioni:

- Le specie, citate da parte del testo, richiedono solo un piccolo ragionamento; ogni specie contiene più esemplari e questi ultimi fanno parte di una sola specie. A questi scopi viene istituita l'entità Specie
- L'informazione forma aggregata non specifica altre informazioni utili, semplicemente le piante fiorite e ad alto fusto sono due cose diverse già descritte dalle entità
- Il trattamento a cui devono essere sottoposte le piante dipende dal tipo di terriccio; pertanto, quest'ultimo attributo è la chiave di Terriccio, di cui fanno parte N specie delle piante ad alto fusto
- Il fatto che ci sia una generalizzazione per Vendita all'ingrosso viene suggerito dal testo perché dice che "*Solo per le vendite all'ingrosso, interessa conoscere l'acquirente*"
- Si citano due tipi di clientela, al che uno penserebbe alla generalizzazione totale da Acquirente al dettaglio e Commerciante all'Ingrosso verso Cliente. Il testo però cita espressamente che solo per i commercianti all'ingrosso si intendono salvare dati e, nel successivo pezzo di testo, specifica pure che solo per la vendita all'ingrosso vogliamo salvare dati.
  Per tutti questi motivi, Vendita si prefigura come entità, Vendita Ingrosso rappresenta una generalizzazione parziale, perché *eventualmente* salviamo i dati utili

Ristrutturazione



Dato che vogliamo minimizzare i valori nulli, idealmente a noi interessa solo capire se la vendita sia all'ingrosso o meno. La generalizzazione può quindi essere risolta togliendo Vendita Ingrosso ed aggiungendo un campo Tipo, che qualifica in che modo la vendita avvenga.
Mantenendo solamente una relazione senza aggiungere il campo Tipo, si risolverebbe la generalizzazione, ma si salverebbero anche i dati di tutte quelle vendite che non sono all'ingrosso.

Dato che i sistemi tradizionali per la gestione delle basi di dati non consentono di rappresentare costrutti quali generalizzazioni, gerarchie, attributi composti o identificatori esterni, risulta necessario ristrutturare lo schema precedente trasformandoli in elementi rappresentabili. In questo caso, nella trasformazione, bisogna solamente rimuovere la generalizzazione Vendita-Vendita Ingrosso.
Per far ciò aggiungiamo un attributo "Tipo" alla classe Vendita e aggiungiamo il seguente vincolo di integrità:
- L'attributo Tipo della classe Vendita può assumere solamente i valori {All'ingrosso, Non all'ingrosso}.

Schema logico:

Terriccio (<u>TipoTerreno</u>)
Specie (<u>NomeSpecie</u>)
Adatto (<u>Specie</u>, Terriccio)
- Adatto.Terriccio -> Terriccio.TipoTerreno
- Adatto.Specie -> Specie.NomeSpecie

PiantaAltoFusto (<u>ID</u>, NomeBotanico, Età, Clima, TipoTerreno, NomeSpecie)
- PiantaAltoFusto.NomeSpecie -> Specie.NomeSpecie

ParteAltoFusto (<u>Vendita</u>, <u>PiantaAltoFusto</u>, Prezzo)
- ParteAltoFusto.Vendita -> Vendita.ID
- ParteAltoFusto.PiantaAltoFusto -> PiantaAltoFusto.ID

Vendita (<u>ID</u>, Data, Tipo, PIVANegoziante) (PIVANegoziante è chiave esterna a Negoziante all'ingrosso partita IVA)
- PIVA può essere NULL

ParteFiorita (<u>Vendita</u>, <u>PiantaFiorita</u>, Numero)
- ParteFiorita.Vendita -> Vendita.ID
- ParteFiorita. PiantaFiorita -> PiantaFiorita.NomeBotanico

PiantaFiorita (<u>NomeBotanico</u>, NumEsemplari, CostoEsemplare, FornituraDa, FornituraA, DurataFioritura)
Negoziante Ingrosso (<u>PIVA</u>, CF, Nome, Cognome)

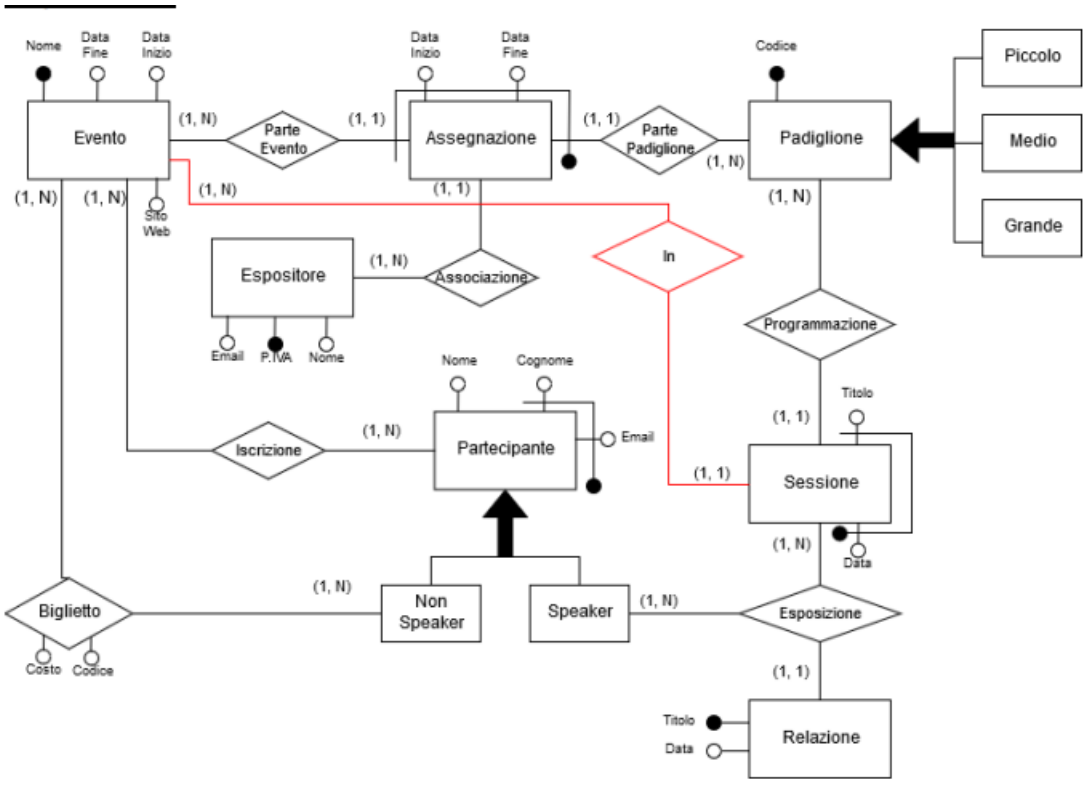
Un’azienda che gestisce gli eventi di uno spazio di fiera vuole progettare una base di dati per la memorizzazione delle informazioni di suo interesse.

L’azienda ha il compito di gestire tutti gli eventi che sono organizzati nello spazio di fiera. Questi eventi sono caratterizzati da un nome, dalla durata dell’evento e, se disponibile, dal sito web dell’evento.

L’evento si svolge nei padiglioni dello spazio di fiera. È di interesse memorizzare quali padiglioni sono occupati in quali giorni. Inoltre, i padiglioni, oltre ad essere identificati mediante un codice alfanumerico, sono classificabili in padiglioni grandi, medi e piccoli. Ogni padiglione ha associato un espositore, di cui interessa il nome, la partita iva, ed il riferimento ad una persona di contatto (e-mail).

Il padiglione ha un programma delle giornate in cui si svolge l’evento che risulta organizzato in sessioni. A ciascuna sessione è associato un titolo, la giornata in cui si svolge e un insieme di speaker, di cui interessa nome, cognome, e-mail, oltre che titolo e durata della relazione che dovranno esporre.

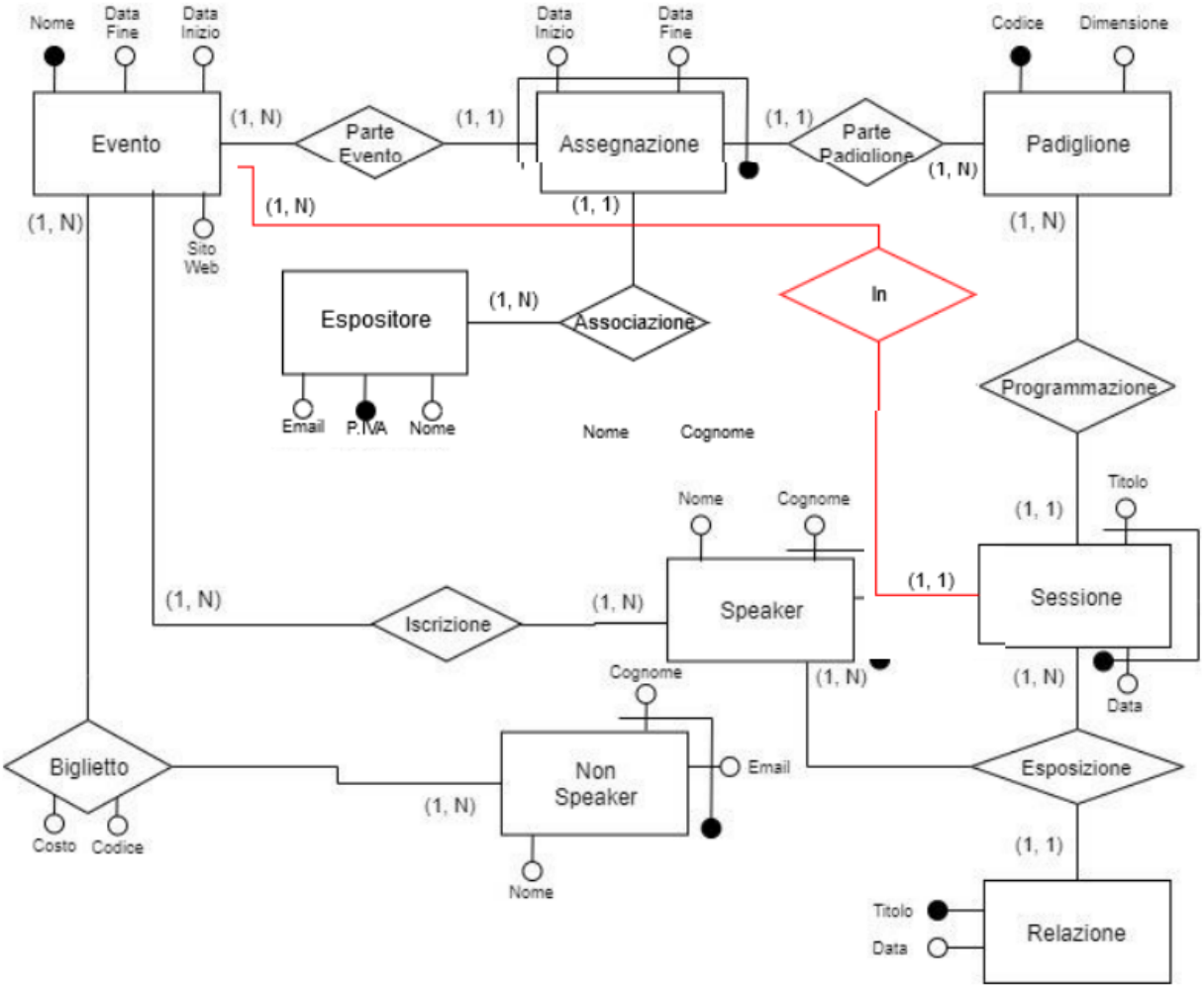
Si noti che uno speaker può effettuare più di una relazione all’interno dello stesso evento. I partecipanti all’evento devono iscriversi fornendo i loro dati anagrafici (nome, cognome, e-mail). Anche gli speaker devono iscriversi all’evento. Tutti i partecipanti tranne gli speaker devono poi pagare un biglietto di iscrizione caratterizzato da un codice identificativo ed un costo.



L’uso del colore rosso è per rendere la relazione “In” più visibile a causa delle sovrapposizioni.

Il diagramma ER possiede 2 generalizzazioni:

- Rimuoviamo le classi Piccolo, Medio, Grande e aggiungiamo un nuovo attributo Dimensione alla classe Padiglione. Aggiungiamo quindi il vincolo che tale attributo può assumere solamente i valori {Piccolo, Medio, Grande}.
- Le classi Speaker e Non-Speaker ereditano gli attributi e le relazioni della classe Partecipante. Tuttavia, Non-Speaker era già in relazione con Evento, quindi nessuna relazione viene aggiunta.



Schema logico:

Evento (Nome, DataFine, DataInizio, SitoWeb)
Assegnazione (DataInizio, DataFine, Padiglione, Nome-Evento, PIVA_Espositore,)

- Assegnazione.Padiglione -> Padiglione.Codice
- Assegnazione.PIVA_Espositore -> Espositore.PIVA
- Assegnazione.Nome-Evento -> Evento.Nome

Espositore (PIVA, Nome, E-mail)
Padiglione (Codice, Dimensione)
Sessione (Titolo, Data, Padiglione, Evento)

- Sessione.Padiglione -> Padiglione.Codice
- Sessione.Evento -> Evento.Nome

Speaker (Cognome, E-mail, Nome)
Iscrizione (Evento, SpeakerCognome, SpeakerEmail)

- Iscrizione.Evento -> Evento.Nome
- Iscrizione.SpeakerCognome -> Speaker.Cognome
- Iscrizione.SpeakerEmail -> Speaker.E-mail

Biglietto (Evento, Non-SpeakerCognome, Non-SpeakerEmail, Costo, Codice)

- Biglietto.Evento -> Evento.Nome
- Biglietto.(Non-SpeakerCognome, Non-SpeakerEmail) -> Non-Speaker.(Cognome,Email)

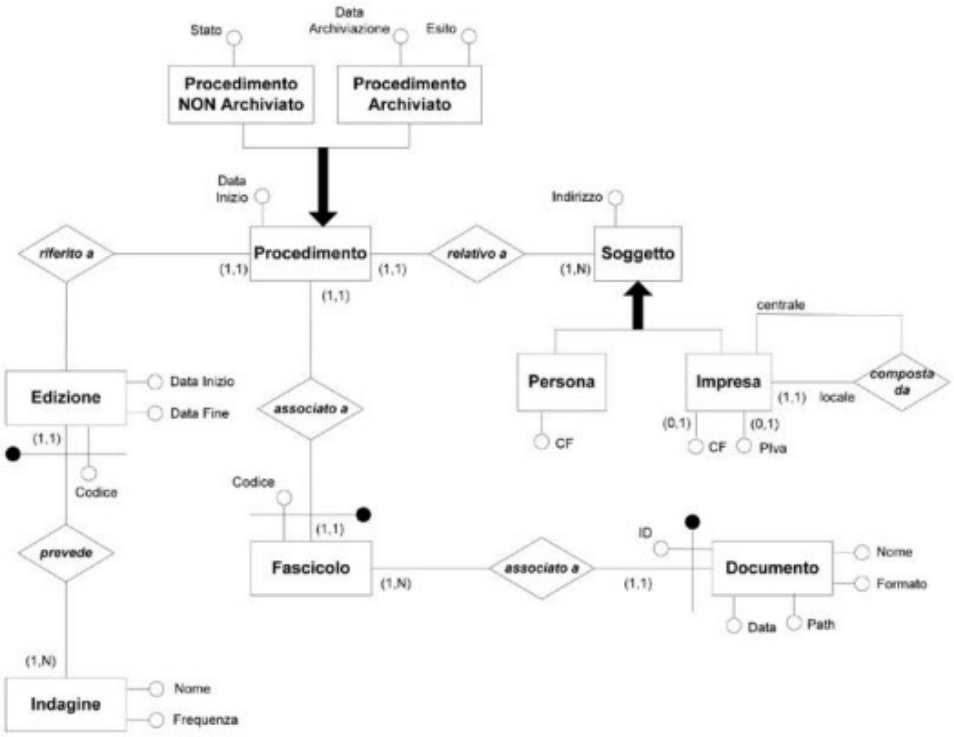
Non-Speaker (Cognome, E-mail, Nome)
Relazione (Titolo, Data, Speaker, Sessione)

Esposizione (SpeakerCognome, SpeakerEmail, SessioneTitolo, SessioneData, Relazione)

- Esposizione.SpeakerCognome -> Speaker.Cognome
- Esposizione.SpeakerEmail -> Speaker.E-mail
- Esposizione.SessioneTitolo -> Sessione.Titolo
- Esposizione.SessioneData -> Sessione.Data
- Esposizione.Relazione -> Relazione.Titolo

Si vuole realizzare una base di dati che gestisca procedimenti sanzionatori nel contesto di rilevazioni statistiche ufficiali di carattere nazionale. Per alcune rilevazioni statistiche ufficiali esiste, infatti, l'obbligo di risposta da parte dei soggetti contattati per la conduzione delle rilevazioni. Qualora il soggetto contattato non risponda al questionario inviatogli, dopo un prefissato intervallo di tempo, ha inizio un procedimento sanzionatorio che consta di due fasi principali: invio della diffida al soggetto non rispondente e, qualora tale soggetto continui ad essere inadempiente (cioè non risponda al questionario), invio della sanzione che il soggetto stesso dovrà pagare. I soggetti possono essere persone fisiche o imprese. Delle persone fisiche interessa memorizzare il codice fiscale, delle imprese il codice fiscale o la partita iva in maniera alternativa. Inoltre, è di interesse l'indirizzo cui il soggetto è contattabile. Si noti che le imprese possono prevedere delle unità locali, ovvero l'impresa si articola secondo una struttura che consiste di un'impresa centrale ed eventualmente di un insieme di imprese "periferiche". Un procedimento viene avviato in relazione alla non risposta ad una specifica edizione di un'indagine. Ogni indagine è caratterizzata da un nome (es. Forze di lavoro), da una frequenza con cui le sue edizioni occorrono (es. trimestrale) e dalle specifiche edizioni che sono occorse (es. primo trimestre 2011).

Le edizioni, che hanno un codice univoco nell'ambito dell'indagine in cui sono svolte, hanno una data di inizio ed una data di fine che caratterizzano l'inizio e la fine della rilevazione sul campo dei dati oggetto dell'indagine. Nell'ambito di un procedimento è prodotto un insieme di documenti che costituisce il fascicolo del procedimento. Un fascicolo ha un codice che lo identifica nell'ambito del procedimento a cui è legato. I documenti, che dispongono di un ID univoco nell'ambito del fascicolo in cui sono redatti, sono rappresentati da un nome, un tipo, una data di produzione e dal path relativo al file cui sono associati. Del procedimento, oltre alle informazioni necessarie a desumere il suo avanzamento, interessa memorizzare la data di inizio e, qualora sia stato archiviato, l'esito della archiviazione (ad esempio archiviato perché il soggetto ha risposto) e la data di archiviazione.

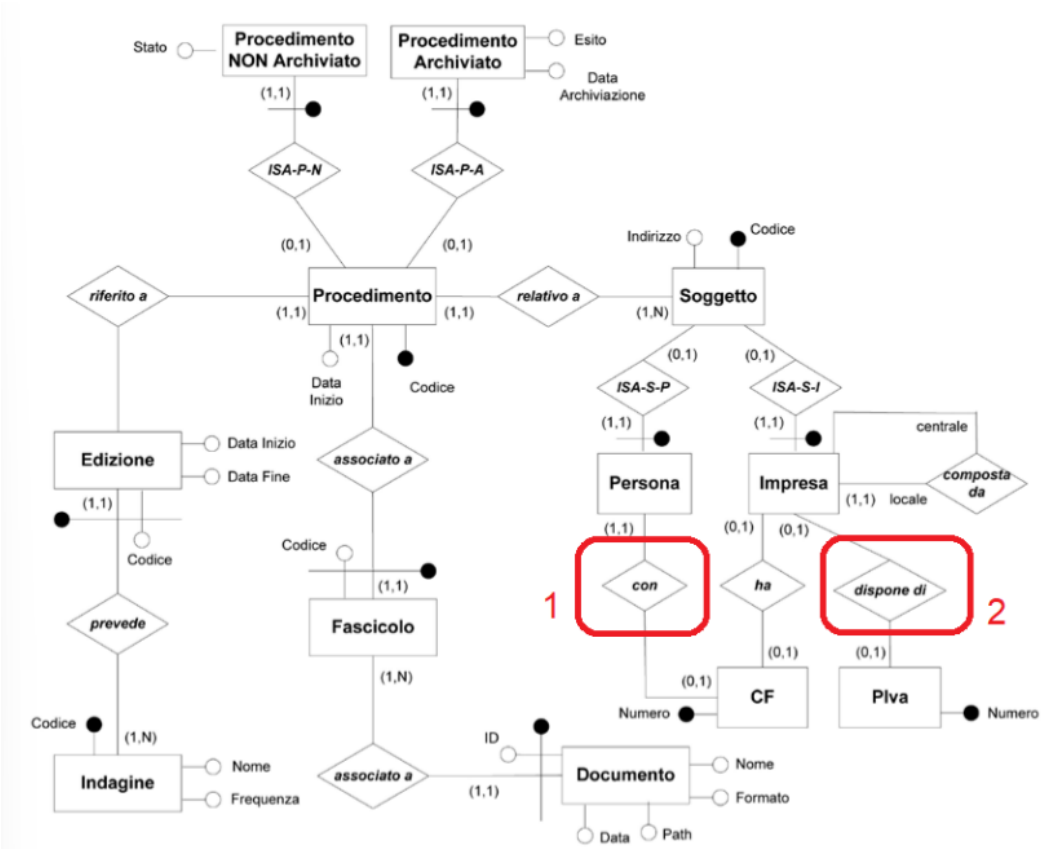


Interessante da esaminare il caso dell'impresa locale e le relative periferiche; dato che non vogliamo memorizzare alcuna informazione in merito a queste singole, si noti che il testo specifica che l'impresa di per sé è locale, dunque costituita da un impresa centrale ed un insieme di periferiche.
La prima idea sarebbe di usare una generalizzazione per rappresentare questo discorso ma non si manterrebbe in alcun modo il riferimento che si ha tra l'impresa locale, di per sé impresa e periferiche; chiaro quindi come si nota una ricorsione, le imprese periferiche dipendono dall'impresa locale e similmente l'impresa locale *potrebbe* avere delle periferiche.
Sono comunque dati irrilevanti al mantenimento dell'informazione del procedimento avviato su di esse; dunque, si struttura la soluzione presentata sopra. Per identificare correttamente l'impresa centrale, viene scelto di aggiungere un campo codice centrale successivamente nello schema logico, distinguendo così tramite codice normale le imprese definite come periferiche.

E ristrutturazione:

Essendo che dobbiamo minimizzare la presenza di valori nulli, allora si sceglie di introdurre delle relazioni tra persona e CF e Impresa con PIVA/CF, mantenendo cardinalità (0,1) da parte di CF/Piva ed (1,1) da parte di Persona ed Impresa.
Similmente per Procedimento, cioè il discorso Archiviato/Non archiviato, mettendo 0,1 da parte di procedimento verso le due sottoentità e (1,1) da parte delle due sottoentità.

Le scelte effettuate nella fase di ristrutturazione hanno avuto come obiettivo quello di evitare la presenza di valori NULL nella base di dati. Questo ha comportato una traduzione "ridondante" delle generalizzazioni, in cui sia l'entità padre che quelle figlie sono state mantenute nello schema, e una traduzione degli attributi con cardinalità (0,1) in nuove entità specifiche. Si noti che quest'ultimo aspetto ha comportato la modifica di uno dei vincoli esterni (quello identificato nel rettangolo num. 2) e l'aggiunta di un nuovo vincolo esterno (quello identificato nel rettangolo num. 1), rispetto a quelli definiti in fase di Progettazione Concettuale.
Inoltre, dato che, per ogni entità, è necessario individuare un campo principale, è stato introdotto un codice, i cui valori sono speciali ed hanno l'unico scopo di identificare le istanze dell'entità, per tutte quelle entità che naturalmente non ne dispongono.



Schema Logico:

Procedimento(Codice, Soggetto, Edizione, Indagine, DataInizio)

- Procedimento.Soggetto --> Soggetto.Codice)
- Procedimento.(Edizione,Indagine) --> Edizione.(Codice, Indagine)

ProcedimentoNonArchiviato(Codice,Stato)

- ProcedimentoNonArchiviato.Codice -->Procedimento.Codice)

ProcedimentoArchiviato(Codice,Esito,DataArchiviazione)

- ProcedimentoArchiviato.Codice -->Procedimento.Codice

Indagine(Codice,Nome,Frequenza)

Edizione(Codice,Indagine,DataInizio,DataFine)

- Edizione.Indagine --> Indagine.Codice

Fascicolo(Codice,Procedimento)

- Fascicolo.Procedimento --> Procedimento.Codice

Documento(ID,Fascicolo,Data,Path,Nome,Formato)

- Documento.Fascicolo -->Fascicolo.Codice

Soggetto(Codice,Indirizzo)

Persona(Codice,CF)

- Persona.Codice -->Soggetto(Codice)
- Persona.CF -->CF.Numero

Impresa (Codice,CodiceCentrale)

- Impresa.Codice -->Soggetto.Codice
- Impresa.CodiceCentrale --> Impresa.Codice

CF(Numero)

PIva(Numero)

ha(CF,Impresa)

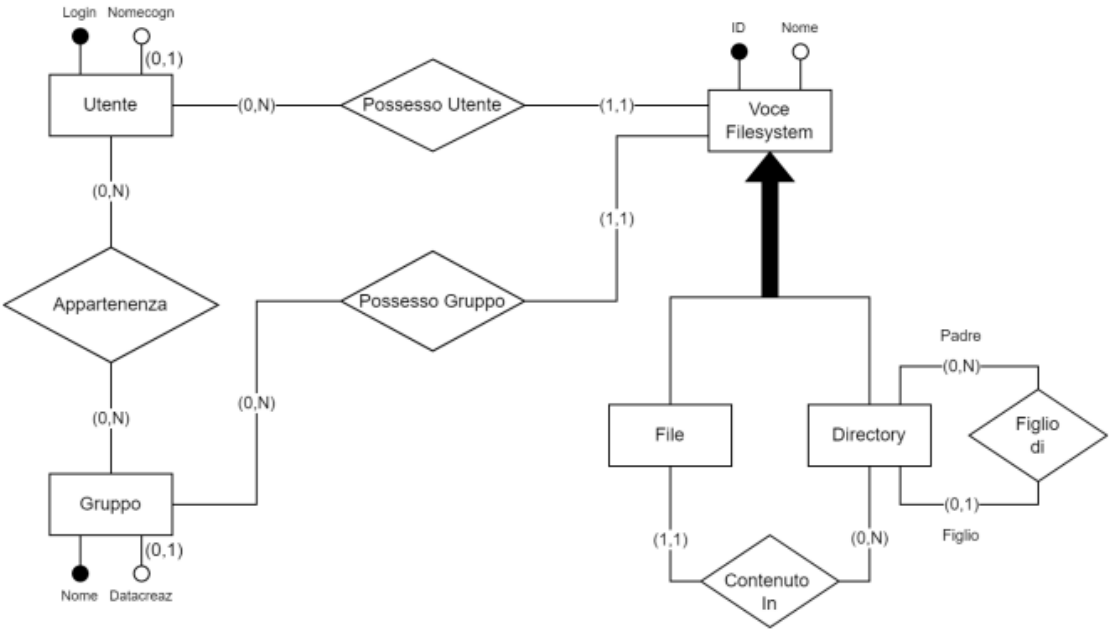
- ha.CF --> CF.Numero

- ha.Impresa --> Impresa.Codice

ImpresadisponeDi(PIva, Impresa)

- disponeDi.PIva --> PIva(Numero)
- disponeDi.Impresa --> Impresa.Codice

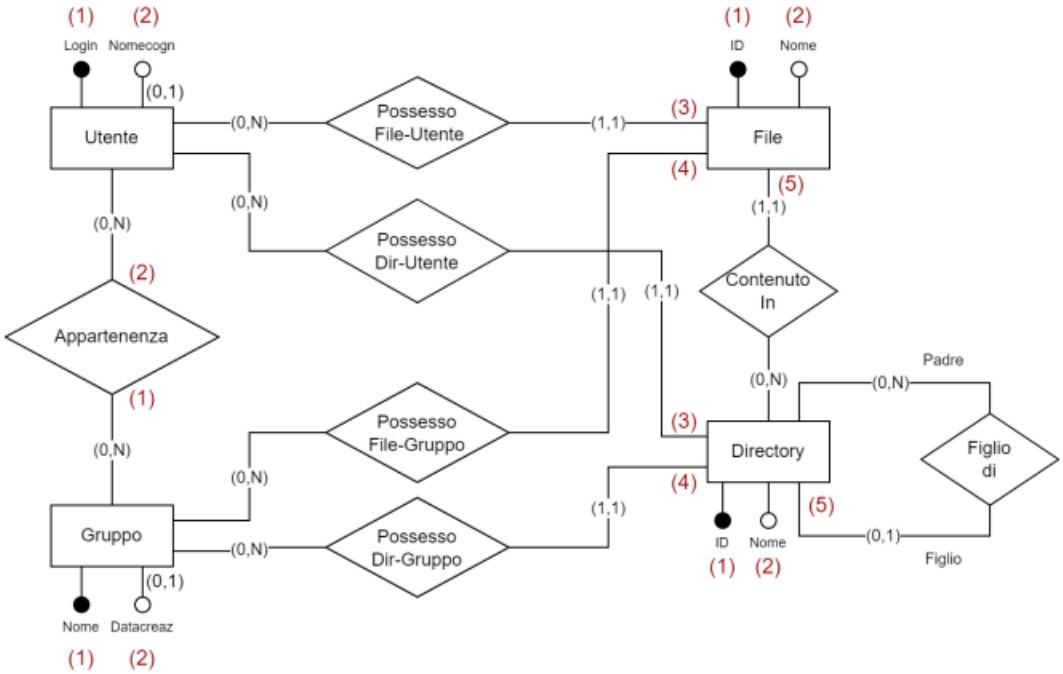
Preso da Laboratorio 4: è evidentemente da ristrutturare:



A tal fine si commenta cosa debba essere fatto:

- Dato che File e Directory hanno entrambi bisogno di essere rappresentati e non ha senso tenere le informazioni di quel genere in Voce Filesystem (portando quindi verso l'alto), allora si può pensare ad introdurre due relazioni per rappresentare il concetto di "is-a".
  Anche questo non ha senso in questo contesto; conviene direttamente portare verso il basso, mantenendo due entità separate File e Directory.
- In tale modo viene introdotta File come entità, collegata con cardinalità (1,1) con Utente e con cardinalità (1,1) verso Gruppo e due relazioni intermedie
- Similmente, viene introdotta Directory come entità, collegata con cardinalità (1,1) con Utente e con cardinalità (1,1) verso Gruppo e due relazioni intermedie
- Le due entità avranno ID come campo chiave e nome come altro attributo

Lo schema ristrutturato quindi è:



I numeri in rosso non rappresentano nulla di utile all'ER, ma l'ordine di inserimento dei campi per entità/relazioni.

Di ogni mobile interessa il "codice unico mobile" (CUM), che identifica il mobile, il numero di giorni impiegati per la sua lavorazione e le parti utilizzate per la sua costruzione. Infatti, ogni mobile è costruito assemblando 2 o più parti e di ogni parte interessa il codice (identificativo), il tipo, ed il tronco usato per produrla (uno ed uno solo).

L'applicazione memorizza le informazioni di vari luoghi geografici (vedi sotto). Di tutti i luoghi geografici interessa il codice (identificativo) e l'area in chilometri quadrati.

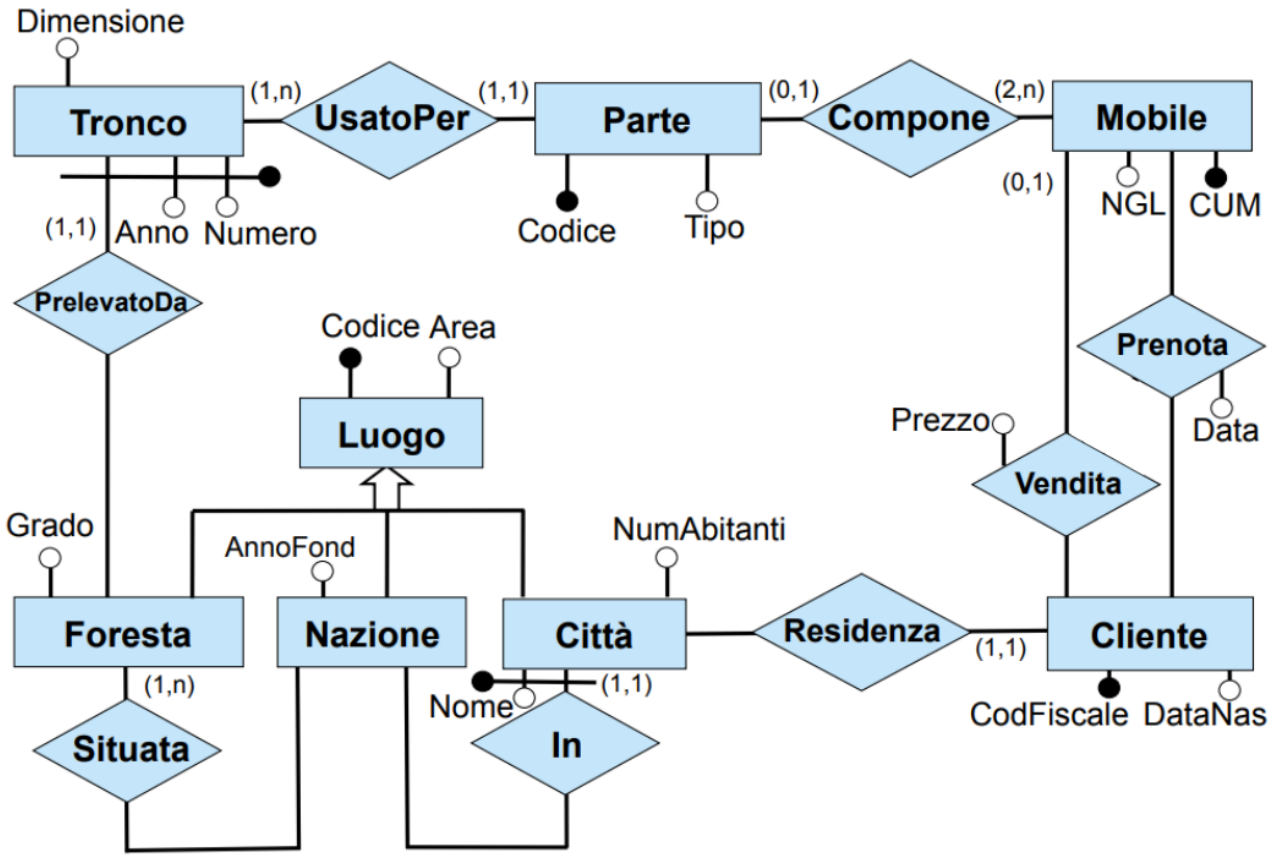
Di ogni mobile interessa sapere chi sono i clienti che lo hanno prenotato ed in quale data lo hanno prenotato. Tra i clienti che hanno prenotato un mobile l'azienda sceglie il cliente al quale vendere il mobile stesso, insieme al prezzo di vendita. Di ogni cliente interessa il codice fiscale (identificativo), la data di nascita e la città di residenza.

Le città sono luoghi geografici di cui, oltre alle proprietà di tutti i luoghi geografici, interessa il numero di abitanti, la nazione in cui si trova ed il nome (unico nell'ambito della nazione in cui si trova).

I tronchi che interessano all'applicazione sono quelli usati per produrre almeno una parte. Ogni tronco viene prelevato da una foresta, ed ha una dimensione. Ad ogni tronco prelevato viene assegnato, usando un contatore, un numero progressivo unico nell'ambito della foresta (all'inizio di ogni anno, il contatore associato ad ogni foresta viene azzerato).

Di ogni foresta, oltre alle proprietà di tutti i luoghi geografici, interessa il grado di inquinamento e le nazioni (almeno una) in cui è situato il suo territorio. Di ogni nazione, oltre alle proprietà di tutti i luoghi geografici, interessa l'anno della sua fondazione.

Schema ER:



I valori nulli quindi saranno:

- Mobile in Parte (perché una parte può non avere un mobile di riferimento)
- Cliente in Mobile (perché un mobile può non avere un cliente di riferimento

Alternativa alla soluzione sotto presente:

- Inserire delle relazioni del tipo “is-a” tra Luogo e le altre Entità. Allo scopo di minimizzare i valori nulli può comunque andare bene la soluzione presentata sotto.

Schema logico

Tronco (Numero, Anno, Dimensione, Nome)
FK:
Tronco.Numero → Parte.Codice_tronco
Tronco.Nome → Foresta.Nome

Parte (Codice_parte, Tipo, Codice_tronco, Mobile)
FK:
Tronco.Codice_tronco → Tronco.Codice
Parte.Mobile → Mobile.CUM

Mobile (CUM, NGI, Cliente, Tronco)
FK:
Mobile.Cliente → Cliente.CF
Mobile.Tronco → Tronco.Codice

Prenotazione (Data_p, Cliente, CUM)
FK:
Prenotazione.CUM → Mobile.CUM
Prenotazione.Cliente → Cliente.CF

Vendita (Cliente, Prezzo, Mobile)
FK:
Vendita.Cliente → Cliente.CF
Vendita.Mobile → Mobile.CUM

Cliente (CF, Data_nascita, Città)
FK:
Cliente.Città → Città.Codice_città

Città (Codice_città, N_abitanti, Nazione)
FK:
Città.Nazione → Nazione.Nome

Nazione (Codice_nazione, Anno, Area)
Collocazione (Nazione, Foresta)
FK:
Collocazione.Nazione → Foresta.Nome
Collocazione.Foresta → Foresta.Nome

Foresta (Grado, Nome)

La generalizzazione parziale viene trattata accorpando l'entità padre nelle figlie, mettendo il codice del luogo e l'area.

Nota: Bisogna mettere anche Area in Città e Foresta

Di ogni blocco di marmo estratto interessa il codice (identificativo), l'anno di estrazione, il peso e la cava da cui è stato estratto.

Per il dominio di estrazione del marmo, è interesse memorizzare i dati di certi luoghi geografici. Ogni luogo geografico ha un codice identificativo e l'area che occupa in chilometri quadrati.

Le cave di interesse sono quelle dalle quali è stato estratto almeno un blocco. Ogni cava è un luogo geografico di cui interessa anche conoscere l'altitudine e la regione (esattamente una) in cui è situato il suo territorio.

Da ogni blocco di marmo si producono almeno una lastra di marmo e di ogni lastra interessa il blocco da cui è stato prodotto (uno ed uno solo), la superficie, il numero (unico nell'ambito del blocco di marmo da cui è stato prodotto), l'eventuale abitazione in cui viene usata.

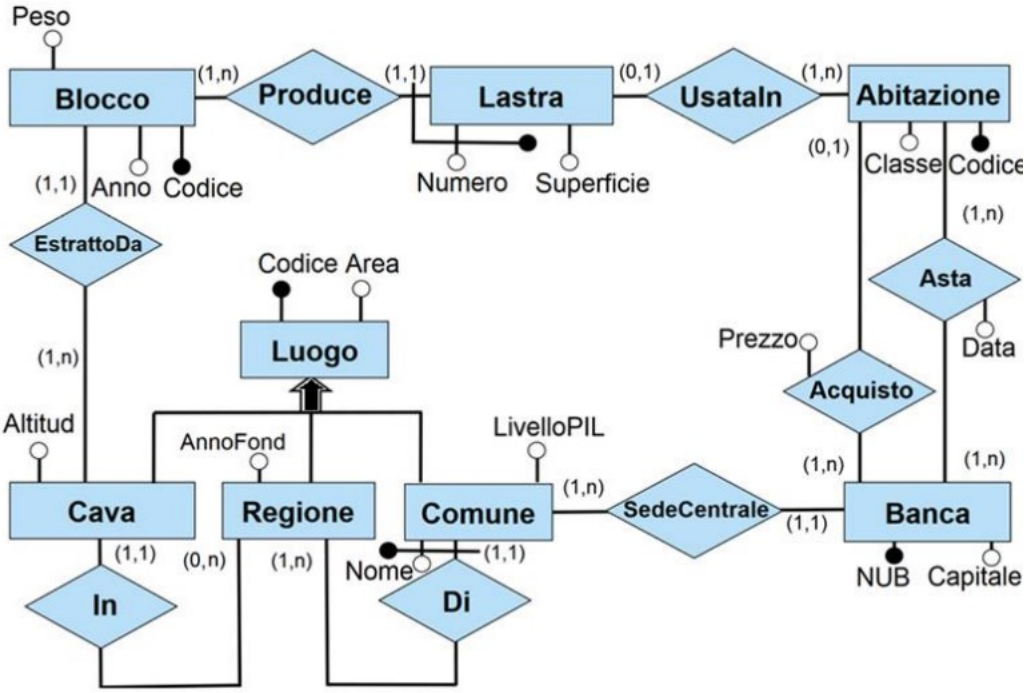
Ogni abitazione che interessa all'applicazione usa almeno una lastra di marmo e di ognuna di tali abitazioni interessa il codice (identificativo), la classe e le eventuali banche che hanno partecipato all'asta per quell'abitazione, con la data di partecipazione della banca all'asta.

Tra le banche che hanno partecipato all'asta per una certa abitazione, dopo tale asta, è di interesse sapere la banca, se esiste, che ha acquistato l'abitazione stessa, con il relativo prezzo di acquisto.

Di ogni banca interessa il "codice unico bancario" (identificativo), il capitale sociale ed il comune in cui si trova la sede centrale.

Di ogni comune, oltre alle proprietà di tutti i luoghi geografici, interessa il livello del PIL, la regione in cui si trova ed il nome (unico nell'ambito della regione in cui si trova).

Di ogni regione, oltre alle proprietà di tutti i luoghi geografici, interessa l'anno della sua fondazione.



L'entità ***Luogo*** viene eliminata, insieme alla generalizzazione. Gli attributi ***Codice*** e ***Area*** vengono replicati in ***Cava***, ***Regione*** e ***Comune***.

Tutti gli attributi seguiti da un asterisco ammettono valori nulli (Nota: all'esame è possibili rappresentare i valori nulli in qualsiasi modo perché non c'è uno standard. L'importante è che sia chiaro quale vengono rappresentati come nulli)

Regione(Codice, Area, AnnoFondazione)
Cava(Codice, Area, Altitudine, CodRegione)
- Foreign key: Cava.CodRegione→Regione.Codice

Comune(Codice, CodRegione, Nome, Area, LivelloPIL)
- Foreign key: Comune.CodRegione→Regione.Codice
- (CodRegione, Nome) è una seconda chiave.

Banca(NUB, Capitale, CodRegSedeCentrale, NomeComuneSedeCentrale)
- Foreign key:
  Banca.(CodRegSedeCentrale, NomeComuneSedeCentrale) →
  Comune(CodRegione, Nome)

Abitazione(Codice, Classe, BancaAcquisto*, PrezzoAcquisto*)
- Foreign key: Abitazione.BancaAcquisto→Banca.NUB

Asta(NUB-Banca,CodiceAbitazione,Data)
- Foreign key: Asta. NUB-Banca →Banca.NUB
- Foreign key: Asta.CodiceAbitazione →Abitazione.Codice

Blocco(Codice,Anno,Peso,CodiceCava)
- Foreign key: Blocco.CodiceCava →Cava.Codice

Lastra(Numero,Blocco,Superficie, AbitazioneUso*)
- Foreign key: Lastra.Blocco →Blocco.Codice
- Foreign key: Lastra.Abitazione →Abitazione.Codice

Definire uno schema Entità-Relazione che descriva i dati di una applicazione relativa all'archivio di un amministratore di condomini, secondo le seguenti specifiche (semplificate rispetto a molte realtà).

- Ogni condominio ha un nome (che lo identifica) e un indirizzo e comprende una o più *scale*, ognuna delle quali comprende un insieme di appartamenti.
- Se il condominio comprende più scale, ad ogni scala sono associati:

  – Un codice (es: scala "A") che la identifica insieme al nome del condominio;
  – Un valore, detto *quota della scala*, che rappresenta, in millesimi, la frazione delle spese del condominio che sono complessivamente di competenza degli appartamenti compresi nella scala.

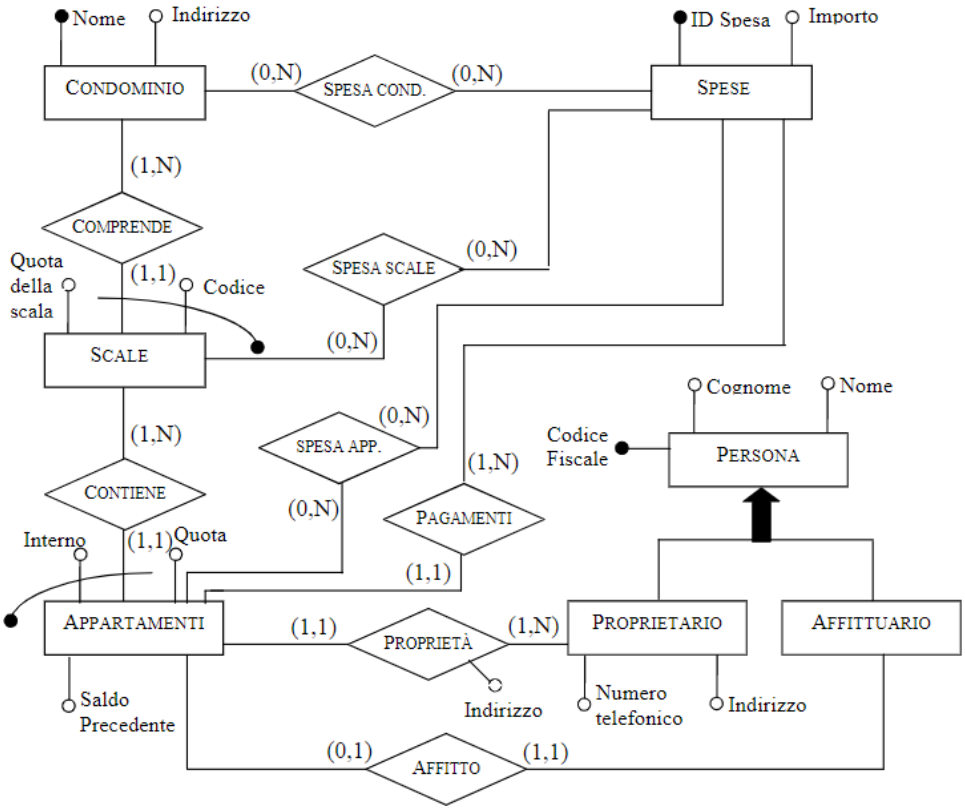
- Ogni appartamento è identificato, nel rispettivo condominio, dalla scala (se esiste) e da un numero (l'*interno*). Ad ogni appartamento è associata una quota (ancora espressa in millesimi), che indica la frazione della spese (della scala) che sono di competenza dell'appartamento.
- Ogni appartamento ha un proprietario per il quale sono di interesse il nome, il cognome, il codice fiscale e l'indirizzo al quale deve essere inviata la corrispondenza relativa all'appartamento. Ogni persona ha un solo codice fiscale, ma potendo essere proprietario di più appartamenti, potrebbe anche avere indirizzi diversi per appartamenti diversi. Di solito, anche chi è proprietario di molti appartamenti ha comunque solo uno o pochi indirizzi. In molti casi, l'indirizzo del proprietario coincide con quello del condominio.
- Per la parte contabile, è necessario tenere traccia delle spese sostenute dal condominio e dei pagamenti effettuati dai proprietari.

  – Ogni spesa è associata ad un intero condominio, oppure ad una scala o ad un singolo appartamento.
  – Ogni pagamento è relativo ad uno e un solo appartamento.

Nella base di dati vengono mantenuti pagamenti e spese relativi all'esercizio finanziario in corso (di durata annuale) mentre gli esercizi precedenti vengono sintetizzati attraverso un singolo valore (il *saldo precedente*) per ciascun appartamento che indica il debito o il credito del proprietario. In ogni istante esiste un *saldo corrente* per ciascun appartamento, definito come somma algebrica del saldo precedente e dei pagamenti (positivi) e delle spese addebitate (negative).

Se e quando lo si ritiene opportuno, introdurre codici identificativi sintetici.

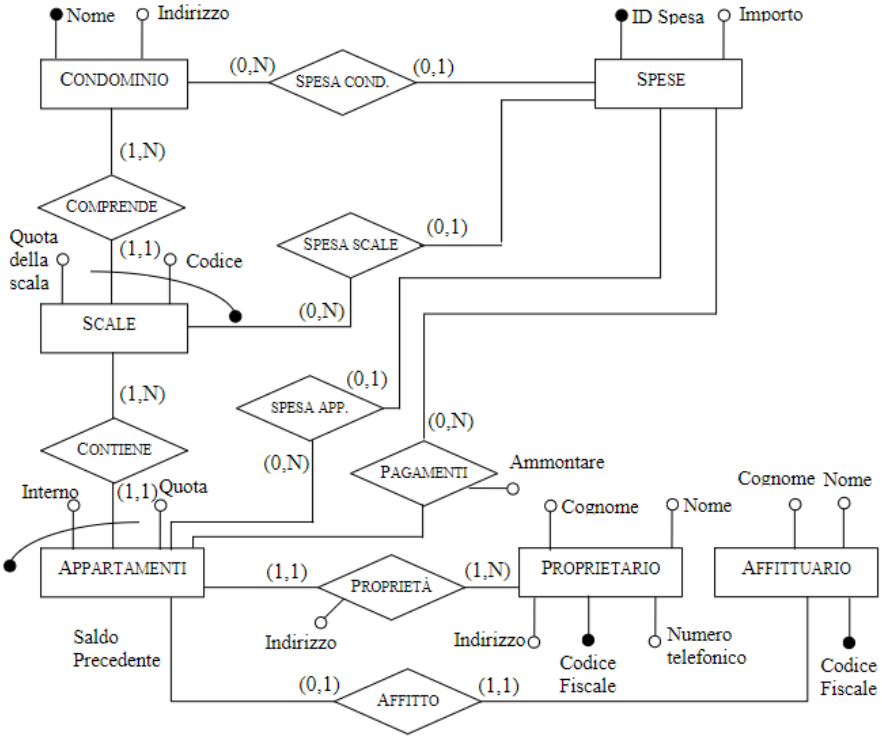
Soluzione:



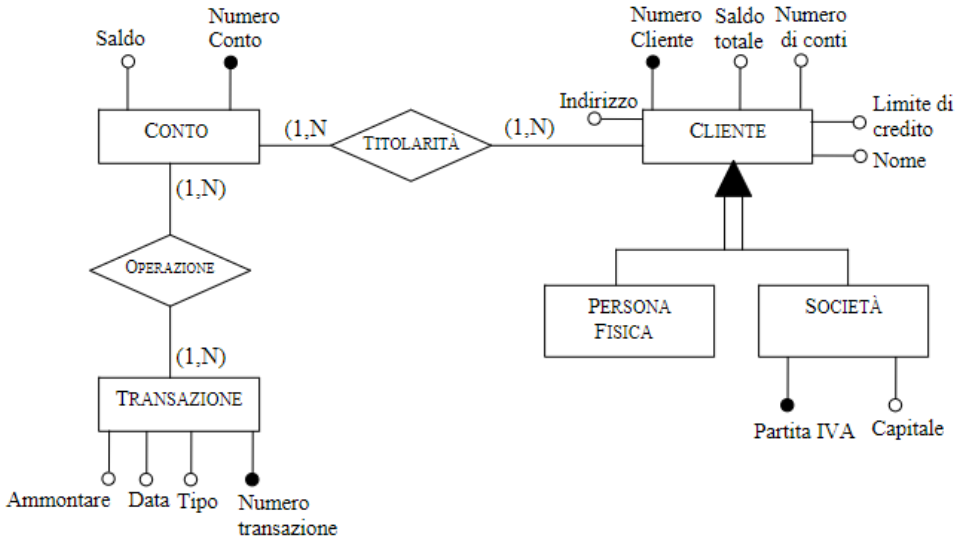
Ristrutturazione di questo schema:

Eliminazione delle gerarchie:

Nello schema è presente una sola gerarchia ed è relativa alle persone ed è totale ed esclusiva, in quanto una persona se è proprietaria di un appartamento non ne è contemporaneamente anche affittuario, quindi si decide di lasciare due entità distinte: l'entità PROPRIETARIO e AFFITTUARIO.



Dato il seguente schema:



La ristrutturazione segue questi principi:

Eliminazione delle gerarchie:

Nello schema è presente una sola gerarchia relativa all'entità CLIENTE, che viene distinto in PERSONA FISICA o SOCIETÀ. L'entità SOCIETÀ ha gli attributi Partita IVA e Capitale che la distinguono. L'unica operazione che fa una distinzione sul tipo di cliente è la numero 10.
Visto lo scarso numero di operazioni e il poco spazio necessario per accorpare le due entità, si decide di accorpare gli attributi Partita IVA e Capitale in Cliente. Sarà l'attributo Partita IVA ad identificare un cliente come società.

Scelta degli identificatori principali:

Gli identificatori sono Numero transazione per l'entità TRANSAZIONE, Numero Conto per l'entità CONTO.
Per quanto riguarda l'entità CLIENTE, l'identificatore è l'attributo Numero cliente; l'attributo Partita Iva identifica le società e se presente deve essere univoco.

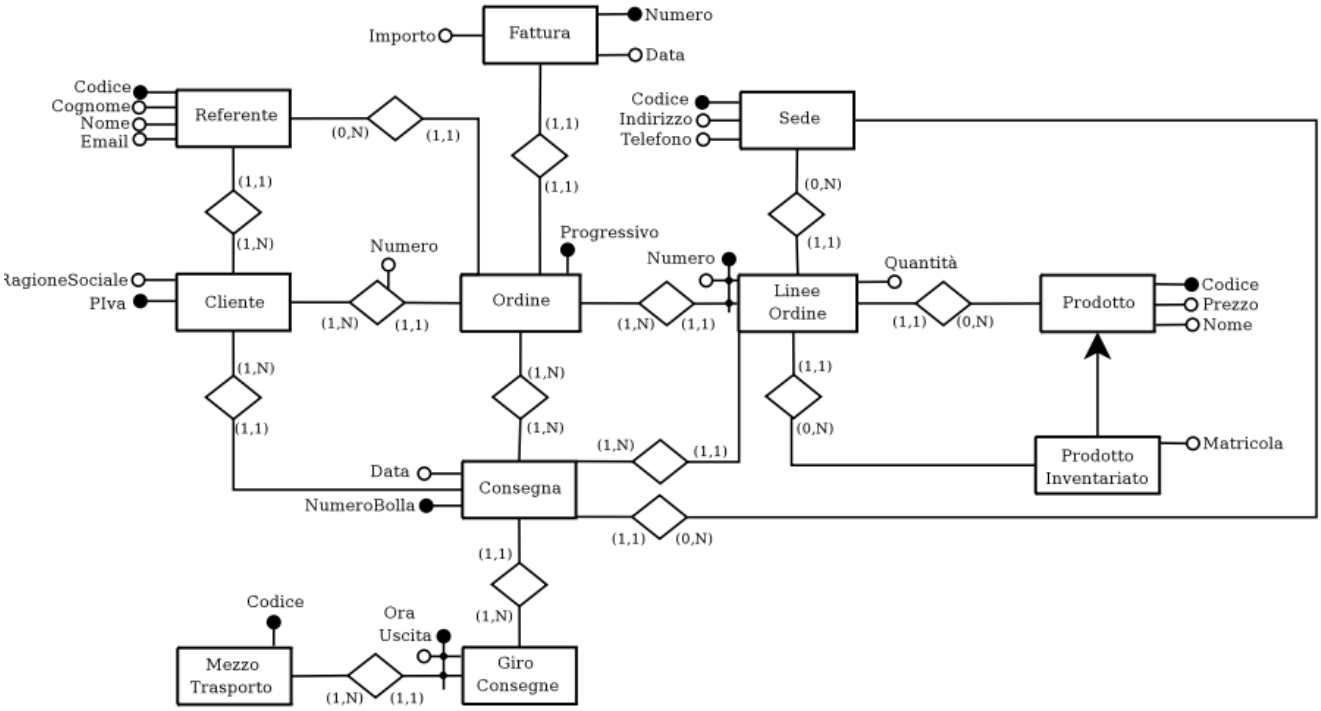
Schema relazionale

TRANSAZIONE(Numero transazione, Tipo, Data, Ammontare)
CONTO(Numero Conto, Saldo)
CLIENTE(Numero cliente, Saldo Totale, Limite di credito, Nome, Indirizzo, Partita IVA*, Capitale*)
OPERAZIONE(Numero conto, Numero transazione)
TITOLARITÀ(Numero conto, Numero cliente)

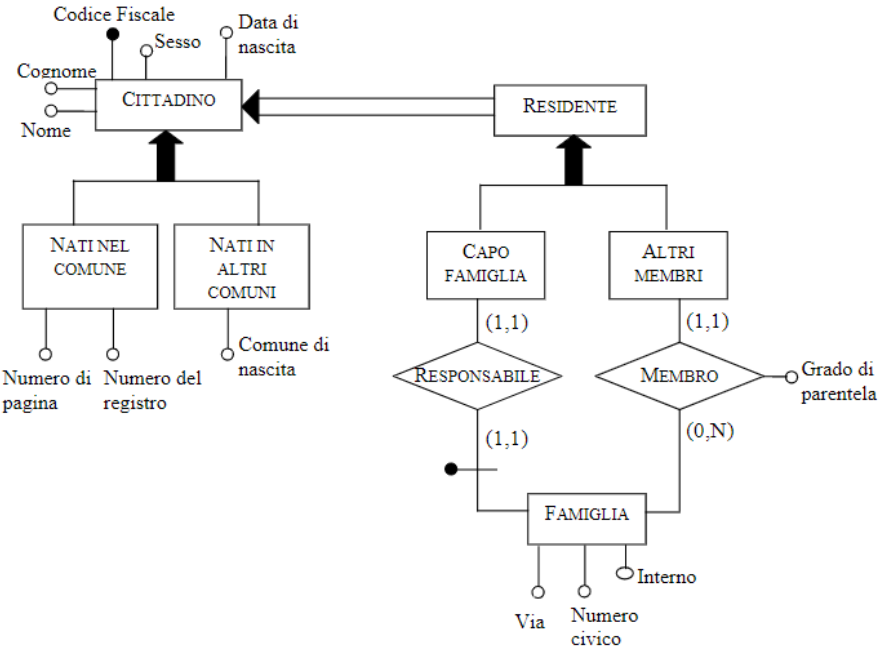
Esercizio 7.17 Definire uno schema E-R che descriva i dati di una applicazione relativa alla gestione ed evasione degli ordini da parte di una azienda, secondo le specifiche elencate di seguito.

- L'azienda riceve gli ordini emessi dai clienti (ognuno dei quali ha numero di partita IVA, che identifica ragione sociale, indirizzo e percentuale di sconto). Ogni ordine ha un numero (attribuito dal cliente), indica il nome di un referente interno del cliente (che può essere lo stesso per tutti gli ordini) e richiede uno o più prodotti, per ciascuno dei quali indica una quantità e una sede di destinazione (in quanto ciascun cliente può, anche nell'ambito di uno stesso ordine, richiedere che i vari prodotti siano consegnati in sedi diverse; ad esempio: "tre calcolatori X386, due stampanti Z322 a via Roma 103 e due calcolatori X343 e una stampante Z320 a Corso Garibaldi 12"). Ad ogni ordine viene assegnato, dall'azienda, all'atto della ricezione, un numero progressivo identificante. Ogni sede di destinazione viene rappresentata da un codice e un indirizzo e non ha correlazione formale con il cliente.
- Gli ordini vengono evasi attraverso consegne, ognuna delle quali è relativa ad un unico cliente e un'unica sede di destinazione, ma può riferirsi a più ordini. Ogni ordine, a sua volta, è soddisfatto attraverso una o più consegne. Per ogni consegna sono rilevanti la data, l'ora e il numero di bolla di accompagnamento. L'azienda ha vari mezzi di trasporto (identificati ognuno da un codice e senza ulteriori proprietà di interesse), ognuno dei quali effettua al più un giro di consegne al giorno, per il quale è di interesse l'ora di uscita dal magazzino.
- Ogni prodotto ha un codice identificante, un nome e un prezzo unitario. I prodotti si dividono in due categorie: inventariabili (per i quali ciascun esemplare ha un numero di matricola di cui si deve tenere traccia nell'ambito della consegna) e di consumo (per i quali è sufficiente far riferimento alle quantità).
- Quando un ordine è stato completamente evaso, viene emessa la fattura, che ha un numero progressivo, una data e un importo.

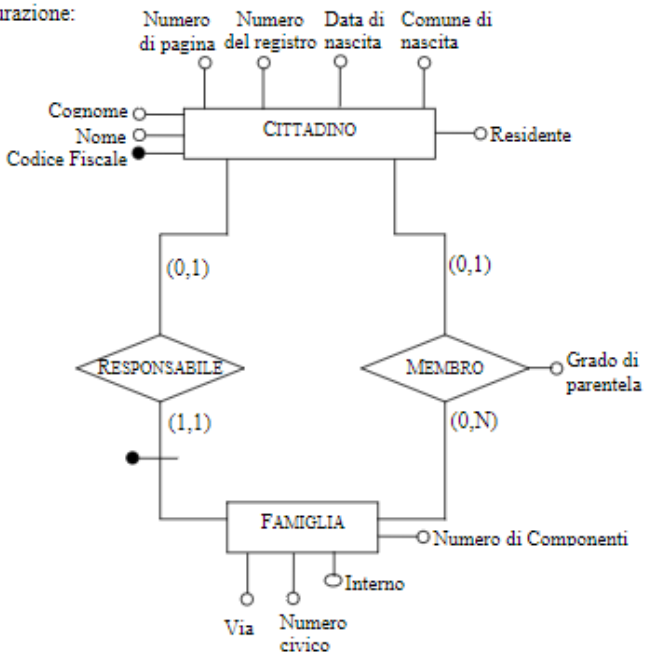
Indicare le cardinalità delle relazioni, (almeno) un identificatore per ciascuna entità e i vincoli non esprimibili per mezzo dello schema. Indicare se è necessario formulare delle ipotesi aggiuntive alle specifiche descritte, senza contraddirle. Limitare le relationship ridondanti che secondo le specifiche potrebbero essere presenti: ad esempio, è evidente che i prodotti sono associati agli ordini e alle consegne (compaiono su vari documenti, ordini, bolle e fatture) ma si richiede di rappresentare tutti i concetti di interesse senza ripeterli. Specificatamente si può pensare di associare i prodotti agli ordini solo inizialmente, per poi associarli alle consegne che, essendo comunque legate agli ordini stessi, permettono di ricostruire l'informazione originaria; per le stesse ragioni, è inopportuno associare i prodotti alle fatture (anche se sulla fattura sono elencati, ma è possibile ricostruire l'elenco per altra via).



Effettuare le ristrutturazioni necessarie in questo schema:



Ristrutturazione:



Traduzioni:

CITTADINO(Codice Fiscale, Cognome, Nome, Numero di pagina, Numero del registro, Data di nascita, Comune di nascita, Residente)

FAMIGLIA(Capo Famiglia, Via, Numero civico, Interno, Numero di Componenti) con vincolo di integrità referenziale tra **Capo Famiglia** e la relazione CITTADINO.

MEMBRO(Cittadino, Famiglia, Grado di parentela)) con vincolo di integrità referenziale tra **Cittadino** e la relazione CITTADINO e tra **Famiglia** e la relazione FAMIGLIA.