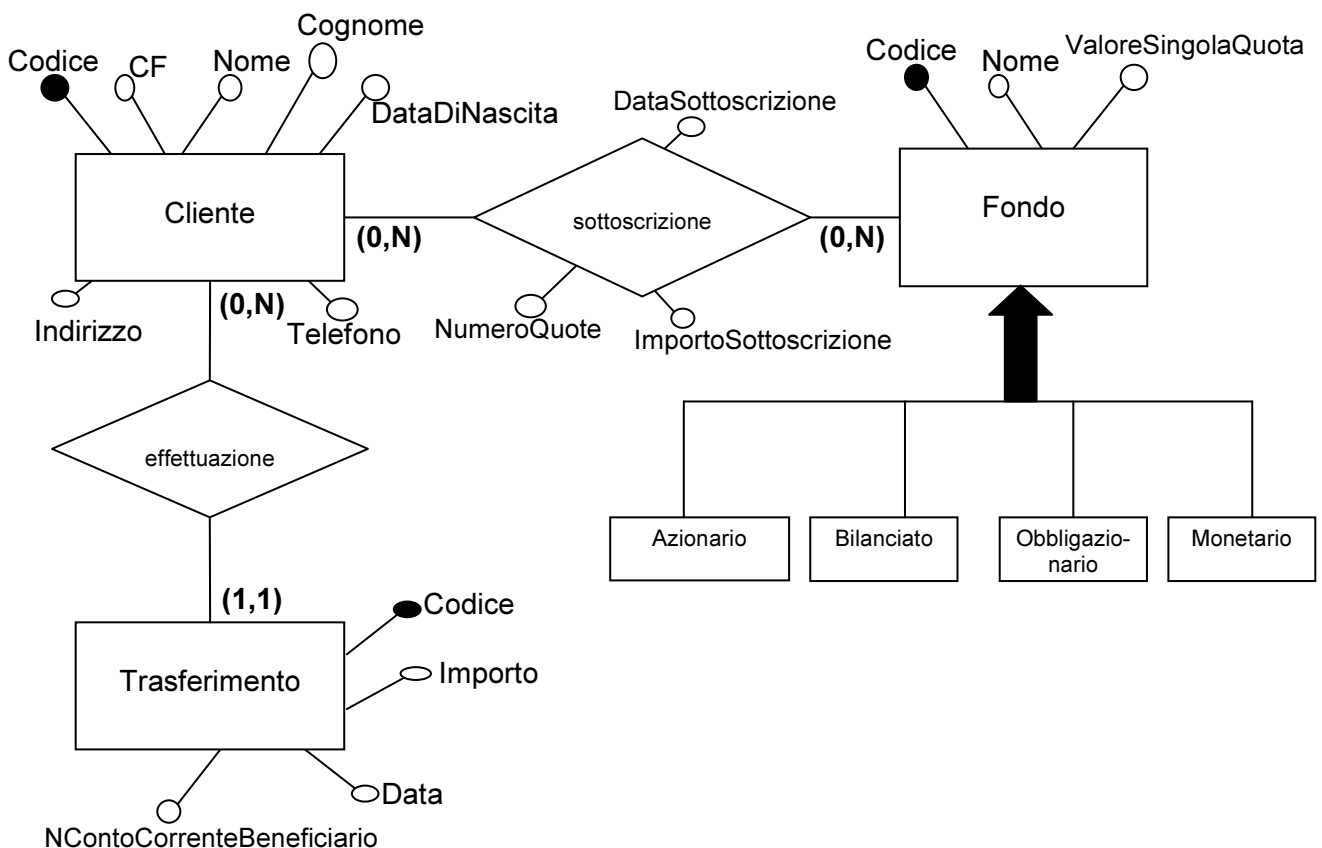


Esercizio 2

La banca Newbank molto attiva nel settore del risparmio gestito, ha deciso di dotarsi di un Sistema Informativo per la gestione dei suoi Clienti. Componente essenziale del Sistema Informativo sarà la Base di Dati *GestioneFondi*.

La progettazione della Base di Dati *GestioneFondi* è iniziata con un'attenta fase di **cattura dei requisiti**. In particolare si è convenuto che la Base di Dati dovrà gestire le informazioni relative ad ogni cliente ed ai fondi di investimento disponibili, memorizzare le sottoscrizioni di ciascun cliente ed i trasferimenti di capitale effettuati dal cliente stesso. Tali requisiti sono stati, quindi, formalizzati opportunamente in un glossario dei termini ed in una lista delle operazioni.

Si è successivamente proceduto alla fase di **Progettazione Concettuale** che ha prodotto il seguente diagramma Entità-Relazione(E-R):



Con riferimento al diagramma E-R sopra riportato si eseguano le seguenti attività:

1. Procedere alla fase di **Progettazione Logica** generando lo Schema Logico Relazionale della Base di Dati.

PROGETTAZIONE LOGICA

DATABASE *GestioneFondi*

```
{  
  TABLE Cliente ( Codice char(6), CF char(16) not null, Nome varchar(30) not null, Cognome  
                  varchar(30) not null, DataNascita Date not null, Telefono  
                  varchar(15), Indirizzo varchar(80) not null,  
                  primary key (Codice), unique (CF) );  
  
  TABLE Fondo ( Codice char(4), Nome varchar(30), ValoreSingolaQuota currency not null,  
                  primary key (Codice), TipoFondo smallint not null,  
                  check(TipoFondo>=1 AND TipoFondo<=4), check(ValoreSingolaQuota>0)  
                  );  
  
  TABLE Trasferimento ( Codice char(10), Importo currency not null, DataTrasferimento date  
                          not null, NcontoCorrenteBeneficiario char(12) not null, primary key  
                          (Codice), CodiceCliente char(6), foreign key (CodiceCliente)  
                          references Cliente(Codice),  
                          check (Importo>0) );  
  
  TABLE Sottoscrizione ( CodiceCliente char(6), CodiceFondo char(4), DataSottoscrizione  
                           date not null, NumeroQuote int not null, ImportoSottoscrizione  
                           currency not null, primary key (CodiceCliente, CodiceFondo),  
                           foreign key (CodiceCliente) references Cliente(Codice), foreign key  
                           (CodiceFondo) references Fondo(Codice),  
                           check (NumeroQuote>0 AND ImportoSottoscrizione>0) );  
}
```

Nota1: La generalizzazione che coinvolge l'Entità Fondo è totale (freccia con tratto pieno) ed esclusiva (un fondo, ad esempio, non può essere contemporaneamente monetario ed azionario) ciò giustifica la soluzione adottata (si veda dispensa sulla progettazione logica). Una soluzione più macchinosa sarebbe stata la seguente:

```
TABLE Fondo ( Codice char(4), Nome varchar(30), ValoreSingolaQuota currency not null,  
              primary key (Codice), Azionario bit, Bilanciato bit, Obbligazionario bit,  
              Monetario bit, check((Azionario=1 and Bilanciato=0 and Obbligazionario=0  
              and Monetario=0) OR (Azionario=0 and Bilanciato=1 and  
              Obbligazionario=0 and Monetario=0) OR (Azionario=0 AND Bilanciato=0  
              AND Obbligazionario=1 AND Monetario=0) OR (Azionario=0 and  
              Bilanciato=0 and Obbligazionario=0 and Monetario=1) );
```

Il vincolo di check mi garantisce l'esclusività della generalizzazione.

Nota2: Non essendo specificato altrimenti nella traccia i vincoli di tupla e di dominio sono a discrezione dello studente pertanto vincoli plausibili ma differenti da quelli presenti nella soluzione proposta sono accettabili.