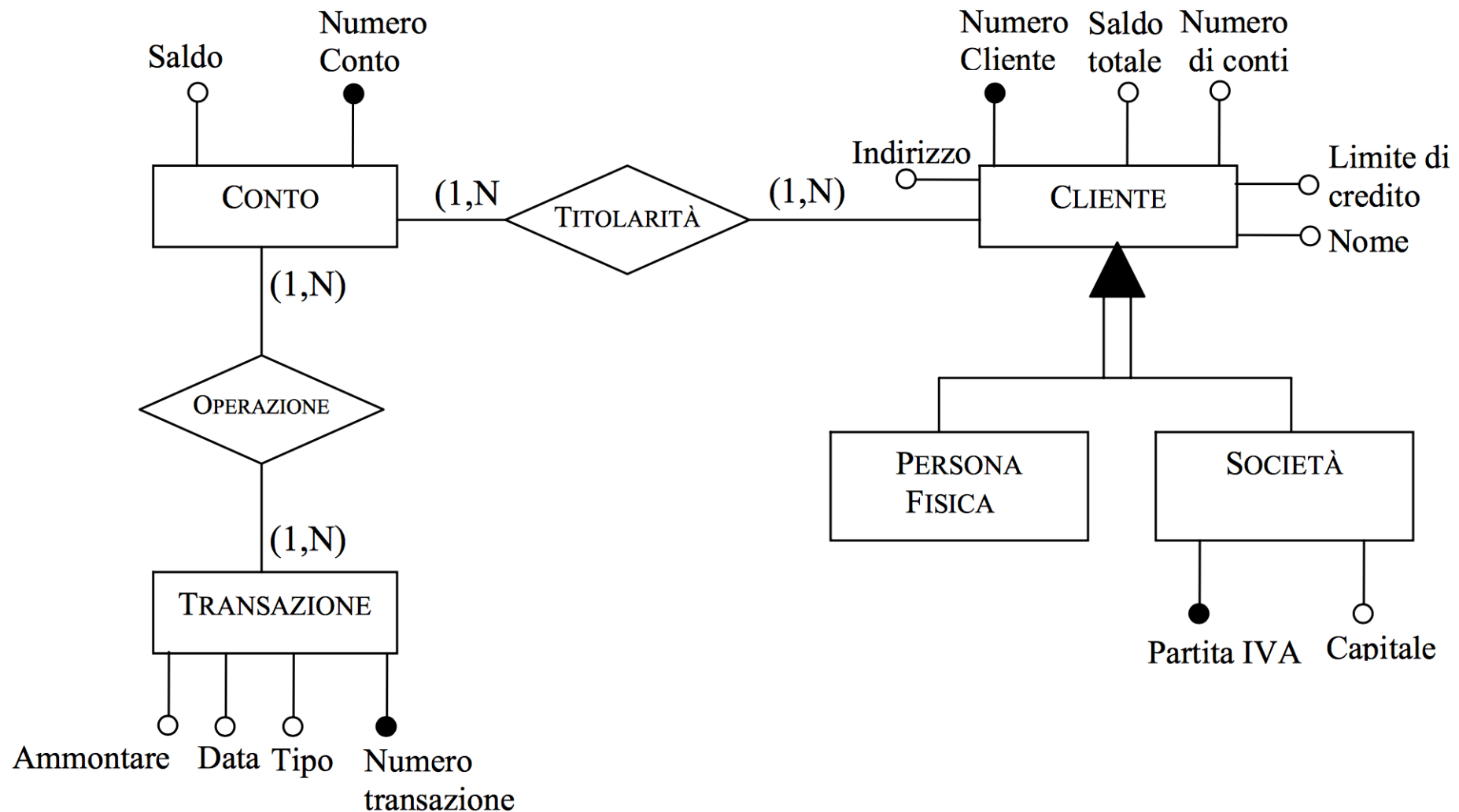


- **Esercizio 1**

- Si consideri lo schema concettuale, che descrive i dati di conti correnti bancari. Un cliente può essere titolare di più conti correnti e che uno stesso conto corrente può essere intestato a diversi clienti.



- Si supponga che su questi dati, sono definite le seguenti operazioni principali:
 - **Operazione 1:** Apri un conto corrente ad un cliente.
 - **Operazione 2:** Leggi il saldo totale di un cliente.
 - **Operazione 3:** Leggi il saldo di un conto.
 - **Operazione 4:** Ritira i soldi da un conto con una transazione allo sportello.
 - **Operazione 5:** Deposita i soldi in un conto con una transazione allo sportello.
 - **Operazione 6:** Mostra le ultime 10 transazioni di un conto.
 - **Operazione 7:** Registra transazione esterna per un conto.
 - **Operazione 8:** Prepara rapporto mensile dei conti.
 - **Operazione 9:** Trova il numero dei conti posseduti da un cliente.
 - **Operazione 10:** Mostra le transazioni degli ultimi 3 mesi dei conti delle società con saldo negativo.

- Si supponga che, in fase operativa, i dati di carico per questa applicazione bancaria siano:

Tavola dei volumi

Concetto	Tipo	Volume
Cliente	E	15000
Conto	E	20000
Transazione	E	600000
Persona Fisica	E	14000
Società	E	1000
Titolarità	R	30000
Operazione	R	800000

Tavola delle operazioni

Operazione	Tipo	Frequenza
Op. 1	I	100/giorno
Op. 2	I	2000/giorno
Op. 3	I	1000/giorno
Op. 4	I	2000/giorno
Op. 5	I	1000/giorno
Op. 6	I	200/giorno
Op. 7	B	1500/giorno
Op. 8	B	1/mese
Op. 9	B	75/giorno
Op. 10	I	20/giorno

- Effettuare la fase di progettazione logica sullo schema E-R tenendo conto dei dati forniti. Nella fase di ristrutturazione si tenga conto del fatto che sullo schema esistono due ridondanze:
 - Gli attributi **Saldo Totale** e **Numero di Conti** dell'entità CLIENTE. Essi possono infatti essere derivati dall'associazione TITOLARITÀ e dall'entità CONTO.

- **Analisi delle ridondanze:**
 - Nello schema esistono 2 dati ridondanti: **Saldo totale** e **Numero di Conti**.
- Saldo totale:
 - Ipotizzando che l'attributo saldo totale sia di tipo float (32 bit e quindi 4 byte per ogni occorrenza), l'utilizzo di questo dato richiederebbe 4×15.000 byte, con un utilizzo di memoria pari a 60 KB.
- Le operazioni coinvolte con questo dato sono la 2, la 4, la 5, la 7 e la 8.
- Si procede analizzando il costo per ognuna di queste operazioni, non conteggiando l'operazione 8 che viene svolta in batch una sola volta al mese.
- Con dato ridondante
 - Per l'operazione 2 abbiamo: 1×2.000 accessi in lettura = 2.000 accessi al giorno
 - Per l'operazione 4 abbiamo: 1×2.000 accessi in lettura (leggo il saldo totale) + 4×2.000 accessi in scrittura (scrivo il nuovo saldo) = 10.000 accessi al giorno (18k accessi in lettura)
 - Per l'operazione 5 abbiamo: 1×1.000 accessi in lettura (leggo il saldo totale) + 4×1.000 accessi in scrittura (scrivo il nuovo saldo) = 5.000 accessi al giorno (9k accessi in lettura).
 - Per l'operazione 7 abbiamo: 1×1.500 accessi in lettura (leggo il saldo totale) + 4×1.500 accessi in scrittura (scrivo il nuovo saldo) = 7.500 accessi al giorno (13.5k in lettura)

- Senza dato ridondante
 - Per l'operazione 2 abbiamo: ipotizzando che la query di ricerca dei conti di un cliente abbia un costo pari a 5 accessi in lettura, moltiplicato per 2.000 operazioni al giorno, ottengo 10.000 accessi in lettura al giorno.
 - Per l'operazione 4 abbiamo: 14k
 - Per l'operazione 5 abbiamo: 7k
 - Per l'operazione 7 abbiamo: 10.5k
- In conclusione, il dato ridondante ho 42.500 accessi, mentre senza il dato ridondante ho 41.500 accessi al giorno. Il dato ridondante mi fa risparmiare 9.000 accessi a fronte di 60 KB di memoria.

- Numero di conti:

- Per quanto riguarda il numero di conti posseduto da un cliente, l'utilizzo di memoria col dato ridondante è di 1 byte per cliente, che equivale a 15 KB di memoria.

Le operazioni coinvolte sono la 1 e la 9.

Anche senza svolgere i conti, si può notare che l'utilizzo del dato è di sole 75 volte al giorno e in modalità batch con l'operazione 9.

Il conseguente miglioramento di efficienza sarà nell'ordine di un migliaio di accessi in meno al giorno. Sarà quindi a discrezione del progettista l'utilizzo o meno del dato.

In questo caso ipotizziamo quindi di non ritenerlo necessario.

- **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)

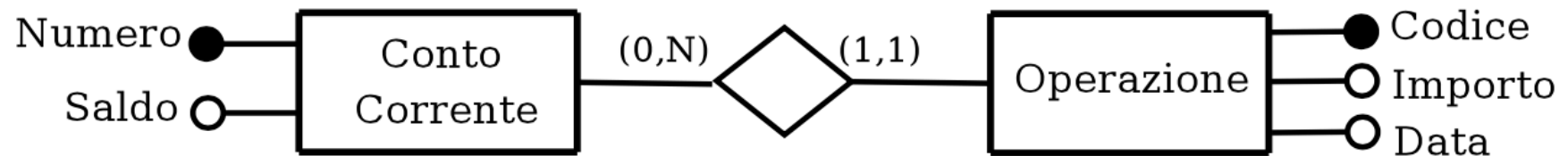
CLIENTE(Numero cliente, Saldo Totale, Limite di credito, Nome, Indirizzo, Partita IVA*, Capitale*)

OPERAZIONE(Numero conto, Numero transazione)

TITOLARITÀ(Numero conto, Numero cliente)

Esercizio

- Si consideri lo schema concettuale



- nel quale l'attributo Saldo di una occorrenza di CONTOCORRENTE è ottenuto come somma dei valori dell'attributo Importo per le occorrenze di OPERAZIONE ad essa correlate tramite la relazione MOVIMENTO.

- Valutare se convenga o meno mantenere la ridondanza, tenendo conto del fatto che le cardinalità delle due entità sono
 - $CC = 2.000$ e $OP = 20.000$
- e che le operazioni più importanti sono:
 - OP1 scrittura di un movimento, con frequenza $f1 = 10$
 - OP2 lettura del saldo con frequenza $f2 = 1000$.