

# Progetto Basi di Dati

Marco Cola 2079237, Davide Martinelli 2077679

## BankDB

### 1. ABSTRACT

Una banca è un'istituzione pubblica oppure privata volta alla raccolta e alla gestione del credito dei suoi clienti, imprese o privati cittadini che siano. Per rappresentare la struttura di una banca si può pensare ai rami di un albero, che si estendono e separano dal tronco principale rimanendo però sempre in contatto con le sue radici; proprio come un albero infatti possiamo pensare alla sede di una banca come le radici, ed i suoi rami, grandi o piccoli che siano, le sue filiali. Ciò che distingue una banca da altre istituzioni e intermediari finanziari è la natura della sua gestione dei fondi, i clienti di una banca infatti possono depositare somme di denaro ed utilizzarle per effettuare transazioni, vale a dire pagamenti, tramite il proprio conto corrente, che rappresenta il principale rapporto tra il cliente e la banca. All'interno di una banca possono esistere diversi tipi di gestione del conto bancario di un cliente, concordati al momento dell'apertura del conto in base alle necessità e all'interesse di quest'ultimo: le due principali tipologie di conto sono il conto corrente e il conto deposito, che si distinguono principalmente per il tipo di operazioni disponibili ed il tasso di interesse offerto. La necessità di avere un database organizzato diventa quindi imperativa per l'istituzione bancaria e le sue filiali, in grado di mantenere un record di tutte le informazioni necessarie e immagazzinare la numerosa quantità di dati generata. L'obiettivo di questo progetto è dunque l'implementazione, la gestione e la documentazione di un database organizzato, che rappresenti gli attributi principali di una banca, come i suoi dipendenti, le sue filiali, i suoi clienti con i relativi conti e le operazioni compiute da essi.

### 2. ANALISI DEI REQUISITI

#### 2.1 Descrizione

Si vuole creare un database per la gestione delle filiali di una banca, con tutti i relativi clienti e i conti a loro associati, i dipendenti e i record delle transazioni effettuate.

In particolare, un **Cliente** è descritto da un codice identificativo, unico per ogni cliente, il Nome, il Cognome (che in caso sia un Azienda rappresenterà il Tipo ad es. S.P.A), la data di nascita, l'indirizzo di residenza: rappresentato da Città, CAP, Via e Numero, il numero di telefono, una mail e il Codice Fiscale.

Ogni Cliente della banca può possedere uno o più Conti per la gestione delle proprie finanze associato alla banca, questo **Conto** è rappresentato dal numero di Conto, univoco e associato al cliente, il Tipo di Conto (Corrente o Deposito), il Saldo attuale, l'IBAN e la Data di Apertura.

Tramite il Conto si possono effettuare delle azioni, identificate come Transazioni. Una **Transazione** nel database è composta dai seguenti attributi: l'ID della Transazione, il Tipo di Transazione (Assegno, Bonifico o Carta), l'Importo, la Data e l'Ora.

Questa Transazione per essere registrata correttamente nel database della banca necessita di un Beneficiario, ovvero chiunque riceva la transazione per conto di un privato o di un'azienda, definendone il Titolo. Il **Beneficiario** è rappresentato da un suo ID, il Nome, Cognome (in caso di Azienda nel campo cognome si mette il tipo di azienda, ad es. S.P.A.), il Titolo (Privato o Azienda), il numero di telefono e la mail.

Ogni elemento sovraelencato può e deve essere ricollegato ad una sede della banca, chiamata Filiale. Una **Filiale** è rappresentata nel database da un ID Filiale, l'indirizzo: contenente gli attributi Città, CAP, Via e Numero, un numero di telefono, l'email e l'orario di Apertura.

Per la rappresentazione dell'astrazione dell'istituzione finanziaria nel database verrà utilizzato il termine **Banca** (ad es. UniCredit), i quali attributi principali sono l'ID Banca, il Nome della Banca, il codice bancario (ABI), l'indirizzo principale della Banca, rappresentato da Città, CAP, Via e Numero, il numero di telefono, la mail e il Tipo di Banca (Popolare oppure Nazionale).

All'interno di una Filiale lavorano i Dipendenti. Gli attributi principali di un **Dipendente** sono: l'ID del Dipendente, il Nome, il Cognome, il Ruolo (che può essere Manager, Dirigente, Dirigente della Filiale, Dipendente d'ufficio), il numero di telefono e la mail.

Ogni Dipendente ha stipulato un Contratto con la Banca, questo **Contratto** è rappresentato da un ID Contratto, la Data di Sottoscrizione, il Tipo di Contratto (che può essere indeterminato, determinato o stage), lo stipendio mensile e la Data di Fine del contratto.

Ogni Dipendente può richiedere delle Ferie, che possono essere accettate o meno in seguito ad una richiesta al proprio direttore, le **Ferie** sono quindi caratterizzate da una Data di Inizio, una Data di Fine e lo stato della richiesta (Accettata, Rifiutata o In Attesa).

## 2.2 Glossario

- **IBAN (International Bank Account Number)**: Rappresenta un codice univocamente associato al Cliente e al suo Conto, costituisce un sistema di salvaguardia dei dati dell'account durante le transazioni.

- **CAP (Codice di Avviamento Postale)**: Si tratta di una serie numerica di cinque cifre utilizzato per indicare la zona postale di appartenenza, es. 36100 Vicenza.

- **ABI (Associazione Bancaria Italiana):** L'ABI è un codice numerico identificativo composto da 5 cifre, la cui prima cifra è sempre uno 0, rappresentante l'istituto di credito. Ogni Banca possiede un codice ABI che viene assegnato proprio dall'Associazione Bancaria Italiana (es [03069](#) Intesa San Paolo S.P.A.).
- **CF (Codice Fiscale):** Il codice fiscale in Italia è un codice che serve a identificare in modo univoco le persone fisiche e altri soggetti diversi dalle persone fisiche nei loro rapporti con gli enti e le amministrazioni pubbliche dello Stato italiano.
- **CAB (Codice di Avviamento Bancario):** codice numerico di cinque cifre che identifica la specifica filiale.

### 3. PROGETTAZIONE CONCETTUALE

#### 3.1 Lista Entità

Il Database è formato dalle seguenti tabelle. Se non specificato l'attributo è NOT NULL.

- **Cliente:** rappresenta un cliente della Banca
  - IDCliente: varchar(16)
  - Nome: varchar(30)
  - Cognome: varchar(30)
  - DataNascita: date
  - Indirizzo: attributo composto da: Via (varchar(30)), Numero (int), CAP (varchar(5)), Città (varchar(30))
  - NumeroTel: varchar(12)
  - Email: varchar(60)
  - CF: varchar(16)
- **Conto:** rappresenta un Conto associato al Cliente
  - NumeroConto: varchar(12)
  - Saldo: numeric (10, 2)
  - IBAN: varchar(27)
  - DataApertura: date
- **ContoCorrente:** Rappresenta un tipo di Conto che consente il prelievo ed il deposito del denaro.
- **ContoDeposito:** Rappresenta un tipo di Conto che consente il deposito del denaro ad un tasso di interesse vantaggioso.
- **Transazione:** indica l'azione compiuta da un Cliente tramite il proprio Conto andando a modificarne il saldo
  - IDTransazione: varchar(12)

- Importo: numeric (10, 2)
- DataOra: timestamp
- **Assegno**: tipologia di pagamento mediante un assegno
- **Bonifico**: tipologia di pagamento tramite bonifico bancario nazionale o estero
- **Carta**: tipologia di pagamento mediante carta di debito/credito
- **Beneficiario**: chiunque riceva la transazione
  - IDBeneficiario: varchar(12)
  - Nome: varchar(30)
  - Cognome: varchar(30)
  - NumeroTel: varchar(12)
  - Email: varchar(60)
- **Privato**: beneficiario della transazione privato, ovvero un singolo individuo
- **Azienda**: beneficiario della transazione rappresentato da un Azienda
  - NomeAzienda: varchar(30)
- **Filiale**: sede separata della Banca
  - IDFiliale: varchar(5)
  - Indirizzo: attributo composto da: Via (varchar(30)), Numero (int),  
CAP (varchar(5)), Città (varchar(30))
  - NumeroTel: varchar(12)
  - Email: varchar(60)
  - OrarioApertura: time
- **Banca**: sede effettiva della azienda/istituzione finanziaria
  - IDBanca: varchar(5)
  - CodiceBancario: varchar(5)
  - NomeBanca: varchar(30)
  - Indirizzo: attributo composto da: Via (varchar(30)), Numero (int),  
CAP (varchar(5)), Città (varchar(30))
  - NumeroTel: varchar(12)
  - Email: varchar(60)
- **Nazionale**: Tipo di Banca finalizzato al profitto azionario, generalmente organizzate come Società per Azioni(S.p.A.) che operano a livello nazionale
- **Popolare**: Tipo di Banca che opera come una società cooperativa nelle comunità locali

- **Dipendente:** identifica un Dipendente all'interno della Banca
  - IDDipendente: varchar(5)
  - Nome: varchar(30)
  - Cognome: varchar(30)
  - NumeroTel: varchar(12)
  - Email: varchar(60)
- **Manager:** Gestore di un certo settore aziendale (Retail, Commerciale o Consulenza)
  - Retail: varchar(6)
  - Commerciale: varchar(11)
  - Consulenza: varchar(10)
- **Dirigente:** Dirigente della Banca
- **DirigenteFiliale:** Dirigente della Filiale
- **DipendentiUfficio:** Dipendenti comuni della Banca
  - NumeroDipendenti: int
- **Contratto:** rappresenta un contratto stipulato tra la Banca e un suo Dipendente
  - IDContratto: varchar(16)
  - DataInizio: date
  - StipendioMensile: numeric(6, 2)
  - DataFine: date
- **Indeterminato:** Tipo di contratto con Data di Fine non specificata
- **Determinato:** tipo di contratto con Data di Inizio e Data di Fine specifiche
- **Stage:** offerta a tempo determinato di stage presso l'azienda
- **Ferie:** rappresenta una richiesta di ferie da parte di un Dipendente con Contratto a tempo indeterminato
  - DataInizio: date
  - DataFine: date
  - StatoRichiesta: varchar(10)

### 3.2 Lista Relazioni

- Banca – Contratto: **Stipulazione**
  - Una Banca stipula almeno uno o più contratti (1,N)
  - Un Contratto viene stipulato con una sola Banca (1,1)
- Banca – Dipendente: **Assunzione**

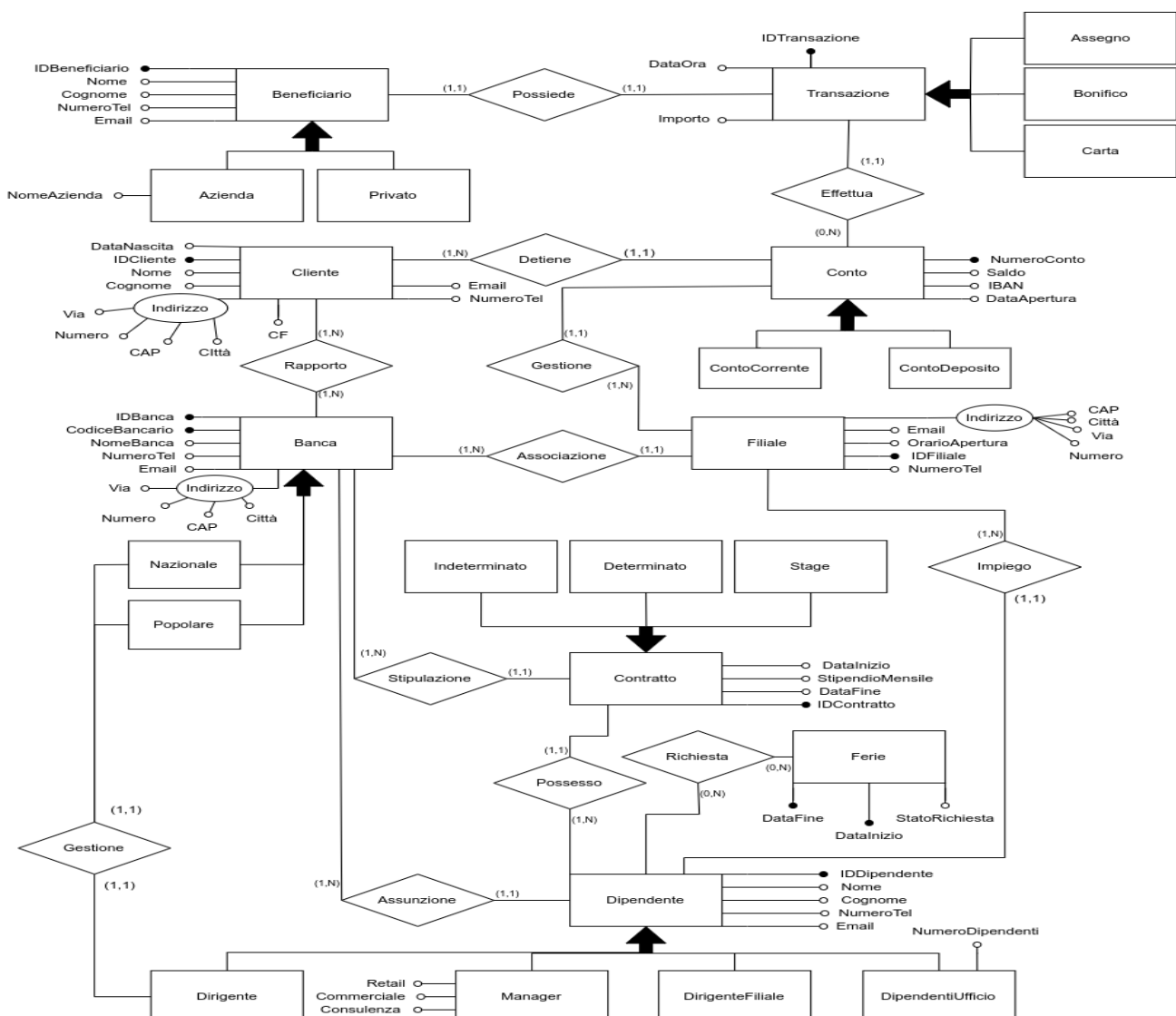
- Una Banca può assumere uno o più Dipendenti (1,N)
- Un Dipendente può essere assunto presso una sola Banca (1,1)
- Banca - Filiale: **Associazione**
  - Una Banca può avere molte Filiali (1,N)
  - Una Filiale appartiene ad un'unica Banca (1,1)
- Banca – Cliente: **Rapporto**
  - Una Banca può possedere uno o più clienti (1,N)
  - Un Cliente può essere gestito da una o più Banche (1,N)
- Dipendente – Contratto: **Possesso**
  - Un Dipendente può possedere uno o più contratti con la Banca (1,N)
  - Un Contratto si riferisce ad un unico Dipendente (1,1)
- Dipendente – Ferie: **Richiesta**
  - Un Dipendente può fare richiesta delle ferie, oppure non fare richiesta (0,N)
  - Un periodo di ferie può essere richiesto da almeno un dipendente, oppure da nessuno (0,N)
- Cliente – Conto: **Detiene**
  - Un Cliente può detenere uno o più conti (1,N)
  - Un Conto è detenuto da un unico Cliente (1,1)
- Conto – Transazione: **Effettua**
  - Un Conto può effettuare una Transazione, oppure non effettuarne alcuna (0,N)
  - Una Transazione appartiene ad un unico Conto (1,1)
- Transazione – Beneficiario: **Possiede**
  - Una Transazione possiede un unico Beneficiario (1,1)
  - Un Beneficiario possiede una sola Transazione (1,1)
- Filiale – Conto: **Gestione**
  - Una Filiale può gestire più Conti (1,N)
  - Un Conto può essere gestito da più Filiali (1,N)
- Filiale – Dipendente: **Impiego**
  - Una Filiale può impiegare uno o più Dipendenti (1,N)
  - Un Dipendente può essere impiegato presso una sola Filiale (1,1)
- Dirigente – Nazionale: **Gestione**
  - Un Dirigente può gestire una sola Banca Nazionale (1,1)

- Una Banca Nazionale è gestita da un solo Dirigente (1,1)
- Dirigente – Popolare: Gestione
  - Un Dirigente controlla una sola Banca Popolare (1,1)
  - Una Banca Popolare ha un solo Dirigente (1,1)

### 3.3 Lista Generalizzazioni

- **Conto** è una generalizzazione totale ed esclusiva di **ContoCorrente** e **ContoDeposito**
- **Transazione** è una generalizzazione totale ed esclusiva di **Assegno**, **Bonifico**, **Carta**
- **Beneficiario** è una generalizzazione totale ed esclusiva di **Privato** e **Azienda**
- **Dipendente** è una generalizzazione totale ed esclusiva di **Manager**, **Dirigente**, **DirigenteFiliale**, **DipendentiUfficio**
- **Banca** è una generalizzazione totale ed esclusiva di **Nazionale** e **Popolare**
- **Contratto** è una generalizzazione totale ed esclusiva di **Indeterminato**, **Determinato** e **Stage**

### 3.4 Schema E-R



## 4. PROGETTAZIONE LOGICA

### 4.1 Analisi delle ridondanze

- L'attributo **Email** dell'entità Banca corrisponde all'attributo Email dell'entità Filiale, in quanto le due entità sono gestite dallo stesso servizio di assistenza online collegato alla sede principale (mentre può cambiare il Numero di Telefono), di conseguenza viene rimosso l'attributo Email dall'entità Filiale.
- La relazione tra Contratto e Banca di **Stipulazione** viene rimossa in quanto il contratto è associato ad un Dipendente, che contiene già le informazioni necessarie sulla Banca di appartenenza.
- L'attributo **Importo** dell'entità Transazione è ridondante in quanto potrebbe essere calcolato guardando le modifiche all'attributo Saldo dell'entità Conto, può essere utilizzato solamente in due operazioni:
  - operazione 1: visualizzazione degli Importi delle transazioni eseguite da un Conto
  - operazione 2: inserimento di una nuova Transazione con relativo importo

Concetto	Tipo	Volume
Transazione	Entità	1000
Effettua	Relazione	1000
Conto	Entità	40

- **operazione 1 (3 volte al giorno):** stampa tutti i dati di Transazione, (inclusi gli Importi, circa  $1000/40 = 25$  per conto).
- **operazione 2 (1000 volte al giorno):** memorizza una nuova Transazione con relativi importi associata ad un Conto.

#### **Presenza di Ridondanza:**

- operazione 1 (x 3 volte al giorno):

Concetto	Costrutto	Accessi	Tipo
Transazione	Entità	1	L

- operazione 2 (x 1000 volte al giorno):

Concetto	Costrutto	Accessi	Tipo
Transazione	Entità	1	L
Transazione	Entità	1	S
Effettua	Relazione	1	S
Conto	Entità	1	S



- **Costi:** Operazione 1 = 3 accessi in lettura.

Operazione 2 = 3000 accessi in scrittura e 1000 accessi in lettura al giorno

- Assumendo costo doppio per gli accessi in scrittura il costo giornaliero è  $3000 \times 2 + 1003$ , ovvero un totale di 7003.

### **Assenza di Ridondanza:**

- operazione 1 (x 3 volte al giorno):

Concetto	Costrutto	Accessi	Tipo
Transazione	Entità	1	L
Effettua	Relazione	1000	L

- operazione 2 ( x 1000 volte al giorno):

Concetto	Costrutto	Accessi	Tipo
Transazione	Entità	1	S
Effettua	Relazione	1	S

- **Costi:** Operazione 1 = 3003 accessi in lettura.

Operazione 2 = 2000 accessi in scrittura

- Assumendo costo doppio per gli accessi in scrittura il costo giornaliero è  $2000 \times 2 + 3003 = 7003$ .

Il costo giornaliero in presenza di ridondanza era di 7003 ovvero rimane uguale, in conclusione l'attributo ridondante non risulta per nulla significativo, viene quindi conservato per consentire una più intuitiva esperienza nell'interrogazione del database.

## **4.2 Eliminazioni delle Generalizzazioni**

- **Conto:** è una generalizzazione totale ed esclusiva, legata alle entità Transazione, Cliente e Filiale. Dal momento che le classi figlie ContoCorrente e ContoDeposito non presentano attributi propri, si possono facilmente accorpare nella classe padre Cliente, aggiungendo a quest'ultima un attributo "**TipoConto**".

- **Transazione:** è una generalizzazione totale ed esclusiva, legata alle entità Beneficiario e Conto, le sue tre classi figlie Assegno, Bonifico, Carta non presentano attributi propri, e tramite l'aggiunta di un attributo "**TipoTransazione**" alla classe padre, possono essere accorpate a quest'ultima, risparmiando memoria.

- **Beneficiario:** è una generalizzazione totale ed esclusiva, legata all'entità Transazione. Le due classi figlie sono Privato e Azienda, entrambe non possiedono relazioni, mentre solamente l'entità Azienda possiede un attributo NomeAzienda, si è deciso quindi di accorpare queste classi figlie alla classe padre, aggiungendo un attributo "**Titolo**". L'attributo NomeAzienda non viene conservato in quanto si può rappresentare con Nome.

- **Dipendente:** è una generalizzazione totale ed esclusiva, in relazione con le entità Banca, Contratto, Filiale e Ferie. Le quattro entità figlie di Dipendente, rispettivamente Manager, Dirigente, DirigenteFiliale, DipendentiUfficio, non presentano attributi o relazioni, tranne Dirigente, in relazione 1 a 1 con i Tipi della banca, DipendentiUfficio, che possiede un attributo NumeroDipendenti e Manager che possiede 3 attributi, rappresentanti i rami aziendali che gestisce; queste classi figlie possono essere accorpate in Dipendente, in quanto la mansione svolta all'interno della Banca può facilmente essere espressa da un attributo "**Ruolo**" nella classe padre Dipendente.
- **Banca:** è una generalizzazione totale ed esclusiva, legata alle entità Cliente, Filiale, Contratto e Dipendente. Le classi figlie Nazionale e Popolare non presentano attributi, sono però in relazione con Dirigente, ora accorpato nell'attributo Ruolo di Dipendente. Si è scelto di accorpare le due classi figlie di Banca in un attributo "**TipoBanca**" all'interno di quest'ultima e assegnare il TipoBanca al Dirigente singolarmente.
- **Contratto:** è una generalizzazione totale ed esclusiva, in relazione con l'entità Dipendente, la classe padre Contratto presenta tre classi figlie: Indeterminato, Determinato e Stage, tutte senza attributi o relazioni, di conseguenza si è scelto di accorparle nella classe base, aggiungendo un attributo "**TipoContratto**", primary key.

### 4.3 Partizionamento/Accorpamento di Entità e Relationship

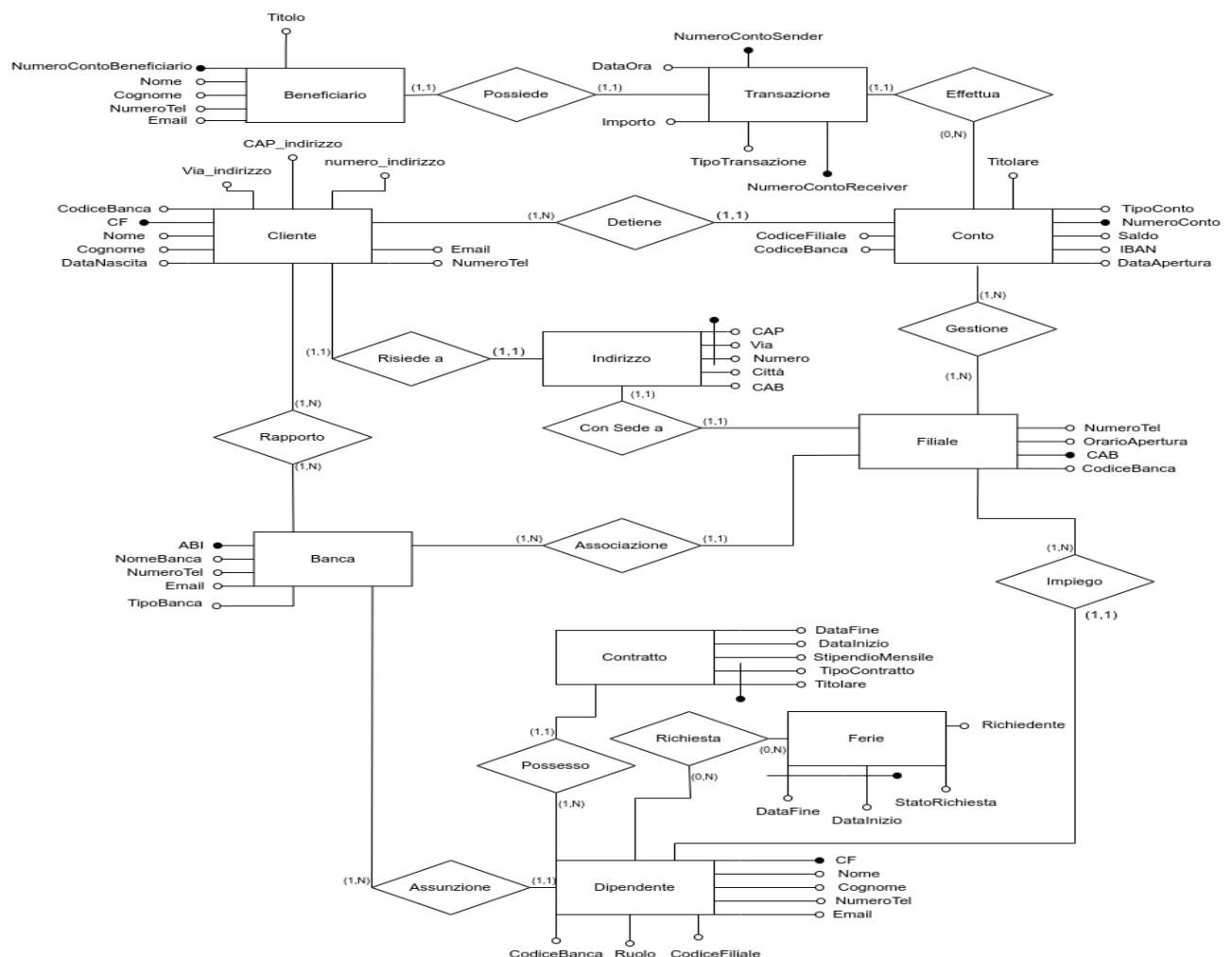
Le entità **Banca**, **Filiale** e **Cliente** possiedono un attributo composto "Indirizzo", per semplificarne l'accesso, viene creata un'entità **Indirizzo**, contenente gli attributi CAP, Città, Via, CAB e Numero, in relazione con le entità interessate Cliente e Filiale, in quanto Banca è solamente un'astrazione del Nome dell'azienda. Vengono aggiunti di conseguenza gli attributi Via\_indirizzo, CAP\_indirizzo e numero\_indirizzo in Cliente come chiavi esterne per consentire il "collegamento" con la nuova entità.

### 4.4 Scelta di identificatori primari e nuovi attributi

- La nuova entità **Indirizzo** avrà come identificatori primari il CAP, la Via e il Numero, in modo da rappresentare al meglio l'indirizzo associato. Viene aggiunto anche l'attributo CAB come chiave esterna per referenziare la Filiale al suo indirizzo.
- All'entità **Contratto** viene aggiunto l'attributo Titolare, che rappresenta il possessore di tale contratto e diventa identificatore primario assieme a TipoContratto, consentendo la rimozione di ID Contratto.
- Nell'entità **Filiale** viene inserito come identificatore primario il CAB, attributo più adatto e utilizzato rispetto ad un troppo comune IDFiliale. Viene aggiunto anche l'attributo CodiceBanca per indicare la Banca di appartenenza della Filiale.
- Nell'entità **Transazione** viene sostituito l'attributo ID Transazione con gli identificatori primari NumeroContoSender e NumeroContoReceiver, per indicare la provenienza e la direzione della transazione, diventa attributo primario anche l'attributo DataOra, per garantire che un Conto possa eseguire più transazioni rispettando i vincoli relazionali.

- Nell'entità **Cliente** viene inserito come nuovo identificatore primario il CF, ovvero il codice fiscale, più adatto e coerente allo scopo rispetto ad IDCliente, che viene rimosso. Viene aggiunto anche l'attributo primario CodiceBanca per indicare a quale banca fa riferimento.
- Nell'entità **Banca** vengono rimossi gli attributi CodiceBanca e IDBanca, sostituiti dal nuovo identificatore primario ABI, che permette un'identificazione più precisa e regolamentata mantenendo lo stesso significato dei precedenti due attributi primari.
- Nell'entità **Beneficiario** viene sostituito IDBeneficiario con NumeroContoBeneficiario, ovvero il numero di conto che riceve la transazione.
- All'entità **Dipendente** infine, per coerenza e comodità viene rimosso l'attributo ID Dipendente e sostituito con il nuovo identificatore primario CF, ovvero il codice fiscale. Vengono aggiunti anche gli attributi CodiceBanca e CodiceFiliale per relazionare Dipendente alla Banca di appartenenza, con la rispettiva Filiale.
- Nell'entità **Ferie**, viene aggiunto l'attributo Richiedente per indicare il Dipendente che fa richiesta di ferie e diventano primary key la data di Inizio e quella di Fine.
- Per concludere, nell'entità **Conto**, vengono aggiunti gli attributi codiceBanca, codiceFiliale e l'attributo Titolare, i primi due per collegare il conto alla rispettiva Filiale e Banca di appartenenza, l'ultimo per relazionare il Conto con il Cliente che lo possiede.

## 4.5 Diagramma E-R ristrutturato



## 4.6 Descrizione schema relazionale

**Vincoli di integrità referenziale:** ( $A \rightarrow B$  indica che A è chiave esterna di B)

**Banca**(ABI, NumeroTel, NomeBanca, TipoBanca, Email)

**Filiale**(CAB, NumeroTel, OrarioApertura, CodiceBanca  $\rightarrow$  Banca.ABI)

**Indirizzo**(CAP, Via, Numero, Città, CAB  $\rightarrow$  Filiale.CAB)

**Cliente**(CF, Nome, Cognome, DataNascita, Email, NumeroTel, CodiceBanca  $\rightarrow$  Banca.ABI, [Via\_indirizzo, CAP\_indirizzo, numero\_indirizzo]  $\rightarrow$  Indirizzo.[via,CAP,numero])

**Conto**(NumeroConto, Saldo, IBAN, DataApertura, TipoConto, Titolare  $\rightarrow$  Cliente.CF, CodiceFiliale  $\rightarrow$  Filiale.CAB, CodiceBanca  $\rightarrow$  Cliente.CodiceBanca)

**Beneficiario**(NumeroContoBeneficiario, Titolo, Nome, Cognome, NumeroTel, Email)

**Transazione**(NumeroContoSender  $\rightarrow$  Conto.NumeroConto, NumeroContoReceiver  $\rightarrow$  Beneficiario.NumeroContoBeneficiario, Importo, DataOra, TipoTransazione)

**Dipendente**(CF, Nome, Cognome, NumeroTel, Email, Ruolo, CodiceBanca  $\rightarrow$  Banca.ABI, CodiceFiliale  $\rightarrow$  Filiale.CAB)

**Contratto**(Titolare  $\rightarrow$  Dipendente.CF, TipoContratto, StipendioMensile, DataInizio, DataFine)

**Ferie**(DataInizio, DataFine, StatoRichiesta, Richiedente  $\rightarrow$  Dipendente.CF)

## 5. QUERY E INDICI

### 5.1 Query

Si riportano di seguito le query presenti all'interno del file codice.c:

1. Query che mostra tutti i conti di una banca specifica e i dettagli dei loro titolari, da terminale permette la scelta della Banca:

```
SELECT c.numeroConto, c.tipoConto, c.saldo, c.dataApertura, cl.nome,
cl.cognome
```

```
FROM conto c
```

```
JOIN cliente cl ON c.titolare = cl.CF
```

```
JOIN banca b ON cl.codiceBanca = b.ABI
```

```
WHERE b.nomeBanca = 'UniCredit';
```

```
Inserisci il numero della tua scelta:
1
Inserisci il nome della banca, scegliendo tra:
Intesa San Paolo, UniCredit, Monte Paschi di Siena, BNL, Banco BPM, Fineco, Banca popolare di Vicenza.
UniCredit
numeroconto    tipoconto    saldo        dataapertura    nome        cognome
925833077389   Corrente    3407592.60    1962-06-02     Pomponio    Combi
616515058712   Deposito    3721213.71    1962-06-02     Pomponio    Combi
514420132508   Deposito    5150334.93    2022-05-30     Fabrizio    Saraceno
118031943539   Deposito    8214384.38    1972-03-04     Luigi       Verdi
286482269098   Corrente    3234459.84    1985-04-14     Luigi       Verdi
```

2. Query che mostra il numero totale di conti per ogni banca, con un raggruppamento per banca:

```
SELECT b.nomeBanca AS Banca, COUNT(c.numeroConto) AS TotaleConti
FROM banca b
JOIN cliente cl ON b.ABI = cl.codiceBanca
JOIN conto c ON cl.CF = c.titolare
GROUP BY b.nomeBanca;
```

```
Inserisci il numero della tua scelta:
2
banca                               totaleconti
Banco BPM                           6
Intesa San Paolo                     2
BNL                                  3
Fineco                              1
UniCredit                           5
Banca popolare di Vicenza           3
Monte Paschi di Siena               3
```

3. Conta il numero di transazioni effettuate dal cliente e calcola il totale degli importi delle transazioni, da terminale permette la scelta del numero minimo di transazioni.

```
SELECT c.CF, c.nome, c.cognome, COUNT(*) AS num_transazioni,
       SUM(t.importo) AS totale_transazioni
```

```
FROM cliente c
```

```
INNER JOIN conto co ON c.CF = co.titolare
```

```
INNER JOIN transazione t ON co.numeroConto = t.numeroContoSender
```

```
GROUP BY c.CF, c.nome, c.cognome
```

```
HAVING COUNT(*) > 2 AND SUM(t.importo) > 1000;
```

```
Inserisci il numero della tua scelta:
3
Inserisci il numero minimo di transazioni, compreso tra 0 e 6: 0
cf      nome      cognome      num_transazioni      totale_transazioni
BRNGIU70A01D206W  Giulia      Bruno        1                    5000.00
MNZFRN85C01H501Z  Franco      Manzi        1                    5000.00
VRDLGI90A01F205X  Luigi       Verdi        6                    11100.00
MSTVRG94C20Y1PFE  Virginia    Mastandrea   2                    1960.00
PIVA-01333550323  Assicurazioni Generali S.p.a.      3                    12800.00
CMBPMN45E51W7YZJ  Pomponio    Combi        2                    2400.00
```

4. Query che mostra il totale delle transazioni per ogni conto inviante (sender) con un saldo superiore a una soglia specifica, selezionabile da terminale.

```
DROP INDEX IF EXISTS idx_saldo_conto;
```

```
CREATE INDEX idx_saldo_conto ON conto(saldo);
```

```
SELECT t.numeroContoSender, COUNT(*) AS TotaleTransazioni
```

```
FROM transazione AS t
```

```
JOIN conto c ON t.numeroContoSender = c.numeroConto
```

```
GROUP BY t.numeroContoSender
```

```
HAVING MAX(c.saldo) > 1000.00;
```

```

Inserisci il numero della tua scelta:
4
Inserisci il saldo minimo desiderato (1000,1MLN,5MLN,7MLN, 9MLN): 1000
numerocontosender    totaletransazioni
497933745241         1
616515058712         1
925833077389         1
118031943539         2
693740880451         1
286482269098         1
788251284763         2
184338094693         3
786629298389         2
823730232912         1

```

5. seleziona il numero di dipendenti per ogni filiale specificando l'indirizzo:

```

SELECT indirizzo.citta, indirizzo.via, indirizzo.numero,
COUNT(dipendente.cf) AS numero_dipendenti, filiale.cab, banca.nomebanca
FROM dipendente
JOIN filiale ON (filiale.cab = dipendente.codicefiliale)
JOIN indirizzo ON (filiale.cab = indirizzo.cab)
JOIN banca ON (filiale.codicebanca = banca.abi)
GROUP BY indirizzo.citta, indirizzo.via, indirizzo.numero, filiale.cab,
banca.nomebanca;

```

```

Inserisci il numero della tua scelta:
5
citta      via      numero  numero_dipendenti  cab      nomebanca
Gottardi nell emilia via Stretto Calbo  52      1      90619    BNL
San Coriolano Piazza Paloma     48      1      75341    Banco BPM
Carmelo sardo via Enzo Moro     37      1      77238    UniCredit
Cattaneo ligure via Vicolo Veneziano 59      3      48203    Banco BPM
Vicenza via Contrada Elvira 74      1      36720    Banca popolare di Vicenza
Sesto Rosina via Contrada Arturo 24      3      92509    Intesa San Paolo
Vicenza via Borgo Naccari 28      1      95268    Banca popolare di Vicenza

```

6. Seleziona lo stipendio per ogni dipendente:

```

SELECT nome, cognome, tipocontratto, ruolo, stipendiomensile
FROM dipendente JOIN contratto ON (contratto.titolare = dipendente.cf)
ORDER BY stipendiomensile DESC;

```

```

Inserisci il numero della tua scelta:
6
nome      cognome      tipocontratto      ruolo      stipendiomensile
Guido      Tarantino     Indeterminato      Manager     2120.00
Annunziata Barberini     Indeterminato      Dirigente Filiale 1900.00
Lina       Opizzi        Determinato        Dirigente Filiale 1750.00
Maria      Giovine       Indeterminato      Manager     1700.00
Luigina    Conte         Indeterminato      Dipendente Ufficio 1520.00
Gioffre    Piazzzi       Determinato        Dipendente Ufficio 1440.00
Giovanni   Rossi         Indeterminato      Manager     1250.00
Dario      Mimun         Determinato        Dirigente Filiale 1200.00
Osvaldo    Galuppi       Stage              Dipendente Ufficio 1100.00
Tonino     Sage         Stage              Dirigente Filiale 1000.00

```

7. Seleziona l'IBAN, la filiale e il nome della banca con relativo indirizzo a cui si devono riferire i clienti:

```

DROP VIEW IF EXISTS cliente_conto;
DROP VIEW IF EXISTS filiale_indirizzo;

```

```

CREATE VIEW cliente_conto AS (SELECT cliente.nome, cliente.cognome,
                                   conto.iban, conto.codiceFiliale, banca.nomeBanca
FROM cliente JOIN conto ON (cliente.cf = conto.titolare AND
                             cliente.codiceBanca = conto.codiceBanca)
JOIN banca ON (cliente.codiceBanca = banca.ABI));

CREATE VIEW filiale_indirizzo AS (SELECT filiale.cab, indirizzo.via,
                                       indirizzo.numero, indirizzo.citta, indirizzo.cap
FROM filiale JOIN indirizzo ON (filiale.cab = indirizzo.cab));

SELECT co.nome, co.cognome, co.iban, co.nomeBanca, fi.citta, fi.via,
       fi.numero
FROM cliente_conto AS co, filiale_indirizzo AS fi
WHERE fi.cab = co.codiceFiliale;

```

Inserisci il numero della tua scelta:

nome	cognome	iban	nomebanca	citta	via	numero
Conad	Società Cooperativa	IT24J0301594573339759179962	BNL	Sesto Edoardo del friuli	Piazza Vittorio	65
Luigi	Zacco	IT73R0503450566943685504536	BNL	Sesto Edoardo del friuli	Piazza Vittorio	65
Angelica	Faggiani	IT5600306968687795917981125	Banco BPM	San Coriolano	Piazza Paloma	48
Mario	Rossi	IT4500301528797786629298389	Banco BPM	San Coriolano	Piazza Paloma	48
Enrico	Pergolesi	IT3660306944499344660121412	Intesa San Paolo	Sesto Rosina	via Contrada Arturo	24
Franco	Manzi	IT24X0306941124497933745241	Intesa San Paolo	Sesto Rosina	via Contrada Arturo	24
Manuel	Lattuada	IT65H0100526867459086496721	Monte Paschi di Siena	Borgo Oreste	via Strada Dallara	78
Assicurazioni Generali	S.p.a.	IT63P0100513660184338094693	Monte Paschi di Siena	Borgo Oreste	via Strada Dallara	78
Fabrizio	Saraceno	IT5T0572826519514420132508	Banca popolare di Vicenza	Vicenza	via Contrada Elvira	74
Giulia	Bruno	IT34V0103030208693740880451	Banca popolare di Vicenza	Vicenza	via Contrada Elvira	74
Virginia	Mastandrea	IT90F0301567812788251284763	Banco BPM	Cattaneo ligure	via Vicolo Veneziano	59
Adele	Iannucci	IT99U0503419871807423430230	Banco BPM	Cattaneo ligure	via Vicolo Veneziano	59
Luigi	Verdi	IT52S0301538598118031943539	Banco BPM	Cattaneo ligure	via Vicolo Veneziano	59
Etta	Verri	IT52L0100533642823730232912	BNL	Bocelli lido	via Contrada Cociarelli	25
Loretta	Salieri	IT87G0503496331993333750876	Banca popolare di Vicenza	Vicenza	via Borgo Naccari	28
Pomponio	Combi	IT88U0503475559925833077389	UniCredit	Carmelo sardo	via Enzo Moro	37
Pomponio	Combi	IT15C0301526584616515058712	UniCredit	Carmelo sardo	via Enzo Moro	37
Luigi	Verdi	IT91I0306988025286482269098	UniCredit	Carmelo sardo	via Enzo Moro	37
Lara	Bianchi	IT55F0301525415000000050876	Fineco	San Temistocle del friuli	Rotonda Sauro	69

## 5.2 Indici

Il Saldo attuale di un Conto è un dato che viene utilizzato molto spesso in lettura e durante l'esecuzione delle query. Si è deciso dunque di aggiungere un indice `idx_saldo_conto` per accelerare la ricerca sui saldi dei conti.

```
CREATE INDEX idx_saldo_conto ON conto(saldo);
```

Nella query numero 4 ad esempio l'utilizzo dell'indice rende più efficiente l'elaborazione della query quando si filtrano i conti in base al saldo, migliorando le prestazioni complessive della query.

## FILE C: VISUALIZZAZIONE DELLE QUERY

Il file `codice.c` contiene il codice in C per l'interrogazione del database tramite le query sopraelencate, viene compilato eseguendo il seguente comando da terminale all'interno della cartella con il file `.c` e le dependencies:

```
gcc codice.c -o codice -I /usr/include/postgresql -lpq
```

il seguente comando genera un eseguibile chiamato `codice`, che andrà eseguito da terminale con il comando `./codice`.