Relazione progetto di Laboratorio

Corso di Basi di Dati

Università degli studi di Udine, A.A. 2024-2025

Daniele De Martin
Enrico Peressin

Massimiliano Di Marco
Michele Vecchiato

**PROGETTAZIONE E IMPLEMENTAZIONE
DI UNA BASE DI DATI PER LA GESTIONE DI UNA BANCA**

Indice

1. Raccolta e analisi dei requisiti

1.1. Richiesta Originale

Si vuole progettare una base di dati di supporto ad alcune delle attività di una banca.
La banca è organizzata in un certo numero di filiali. Ogni filiale si trova in una determinata città ed è identificata univocamente da un nome (si noti che in unc città vi possono essere più filiali). La banca tiene traccia dei risultati (attivi) conseguiti da ciascuna filiale.
Ai clienti della banca è assegnato un codice che li identifica univocamente. La banca tiene traccia del nome del cliente e della sua residenza. I clienti possono possedere uno o più conti e possono chiedere prestitil. Ad un cliente può essere associato un particolare dipendente della banca, che segue personalmente tutte le pratiche del cliente (si tenga presente che non tutti i clienti godono di tale privilegio e che ad un dipendente della banca possono essere associati zero, uno o più clienti).
I dipendenti della banca sono identificati da un codice. La banca memorizza nome e recapito telefonico di ogni dipendente, il nome delele persone a suo carico e il codice dell'eventuale capo. La banca tiene inoltre traccia della data di assunzione di ciascun dipendente e dell'anzianità aziendale di ciascun dipendente (da quanto tempo tale dipendente lavora per la banca).
La banca offre due tipi di conto: conto corrente (con la possibilità di emettere assegni, ma senza interessi) e conto di risparmio (senza la possibilità di emettere assegni, ma con interessi). Un conto èuò essere posseduto congiuntamente da più clienti e un cliente può possedere più conti. Ogni conto è caratterizzato da un numero che lo identifica univocamente. Per ogni conto, la banca tiene traccia del saldo corrente e della data dell'ultima operazione eseguita da ciascuno dei possessori (un'operazione può essere eseguita congiuntamente da più possessori). Ogni conto di risparmio è caratterizzado da un tasso di interesse, mentre ogni conto corrente è caratterizzato da uno scoperto accordato al cliente.
Un prestito (ad esempio, un mutuo) viene emesso da una specifica filiale e può essere attribuito a uno o più clienti congiuntamente. Ogni prestito è identificato univocamente da un codice numerico. Ogni prestito è caratterizzato da un ammontare e da un insieme di rate per la restutuzione del prestito. Ogni rata di un dato prestito è contraddistinta da un numreo d'ordine (prima rata, seconda rata…). Di ogni rata vengono memorizzati anche la data e l'ammontare.

1.2. Analisi dei Requisiti

1.2.1. Assunzioni

Al fine di proseguire con la progettazione concettuale, sono state effettuate le seguenti assunzioni:

- Gli **attivi** sono la somma della liquidità dei conti meno la somma dei prestiti erogati. Sono relativi alla singola filiale.
- Un **cliente** può avere conti in filiali diverse e ogni conto è associato ad una singola filiale.
- I **prestiti** sono legati al conto, non al cliente.
- Un **dipendente** non può gestire se stesso.
- Un **dipendente** può gestire clienti al di fuori della propria filiale e lavora in una sola filiale.
- Il **capo** di un dipendente è l'unico responsabile della filiale in cui il dipendente lavora.
- Nei **conti cointestati** i clienti devono avere lo stesso dipendente (gestore) che li gestisce.
- In caso di **ri-assunzione** di un dipendente, si tiene conto solo dell'ultima assunzione per il calcolo dell'anzianità.
- Tutte le **rate** di un determinato prestito hanno lo stesso ammontare.

1.2.2. Glossario

Per chiarire il significato e le relazioni dei termini chiave definite nei requisiti viene fornito un glossario esplicativo:

Termine	Descrizione	Collegamenti
Filiale	Unità operativa della banca situata in una determinata città. È gestita da un unico capo.	Conto, Dipendente, Capo
Cliente	Persona fisica con almeno un conto aperto nella banca	Conto, Gestore
Conto	Servizio di gestione del denaro che permette diverse operazioni. Può essere esclusivamente corrente o di risparmio	Cliente, Filiale
Conto Corrente	Tipo di conto caratterizzao da uno scoperto	Conto
Conto di risparmio	Tipo di conto caratterizzato da un tasso di interessse	Conto
Gestore	Dipendente che prende in carico le pratiche di uno o più clienti	
Capo	Unico responsabile della filiale presso cui lavora	Dipendente

Tabella 1: *Glossario dei termini chiave*

2. Progettazione Concettuale

2.1. Costruzione dello schema Entità Relazione

L'analisi dei requisiti ha portato alla definizione di un insieme di entità e relazioni che costituiscono il modello concettuale della base di dati.

- L'enittà ***FILIALE*** rappresenta una unità operativa della banca situata in una determinata città. La chiave primaria è il *Nome*, mentre gli altri attributi sono *Città* e *Indirizzo*. Inoltre, per ogni filiale è presente l'attributo derivato *Attivi*, che rappresenta l'ammontare totale della liquidità della filiale e viene calcolato sulla base dei conti, prestiti e rate ad esso associati.

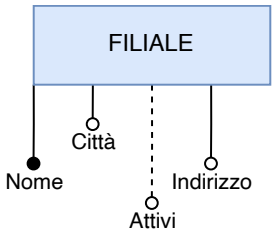

Figura 1: *Entità FILIALE*

- L'entità ***CLIENTE*** rappresenta una persona fisica che ha aperto nella banca almeno un conto. Essa è caratterizzata da un *codice univoco* assegnato dalla banca ad ogni cliente e dal *codice fiscale*, entrambi questi attributi fungono da chiavi primarie in quanto sono univoche per ogni cliente. Gli altri attributi servono per tenere traccia dell'anagrafica del cliente, quali *Nome*, *Cognome*, *numero di Telefono*, *Data di nascita* e *residenza*.

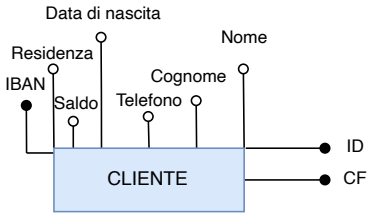

Figura 2: *Entità CLIENTE*

- L'entità ***DIPENDENTE*** è caratterizzata da un codice univoco *ID* che funge da chiave primaria. *Nome*, *Cognome*, *Numero di telefono*, *Data di assunzione* sono gli altri attributi che la descrivono. è stato scelto di tenere traccia dell'anzianità aziendale sulla base della data di assunzione.
  Il capo viene descritto da una specializzazione parziale di *DIPENDENTE*, chiamata *CAPO*.

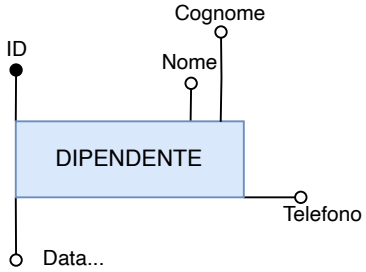

Figura 3: *Entità DIPENDENTE*

- L'entità ***CAPO*** rappresenta il capo di una filiale. Essendo una generalizzazione dell'eneità *DIPENDENTE*, eredita tutti gli attributi di quest'ultima. Un capo è univoco per ogni filiale.

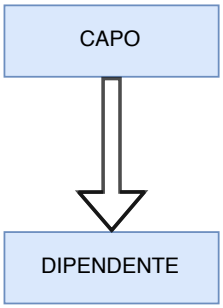

Figura 4: *Entità CAPO*

- L'entità ***CONTO*** serve per identificare un servizio della banca messo a disposizione per il cliente. Ogni entità viene identificata univocamente da un attributo *IBAN* e un attributo *Saldo* tiene traccia dell'ammontare in denaro presente su tale conto. La banca inoltre mette a disposizione due tipi di conto, quindi l'entità Conto è stata specializzata in due sottoentità: *CONTO CORRENTE* e *CONTO DI RISPARMIO*. La specializzazione è totale e disgiunta: l'insieme dei conti correnti e dei conti di risparmio è disgiunto e la loro unione è esattamente l'insieme di tutti i conti all'interno della filiale.

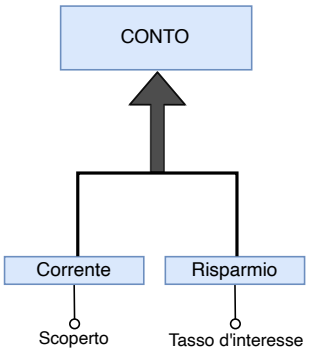

Figura 5: *Entità CONTO*

- L'entità ***CONTO CORRENTE*** è una specializzazione dell'entità *CONTO* pertanto ne eredita tutti gli attributi e tutte le relazioni, la chiave primaria è quindi quella dell'entità *CONTO*. L' attributo che caratterizza *CONTO CORRENTE* è *Scoperto* che indica il valore, concordato tra cliente e banca, di quanto la banca può concedere di debito nei confronti del cliente.
- L'entità ***CONTO DI RISPARMIO*** è una specializzazione dell'entità *CONTO* pertanto ne eredita tutti gli attributi e tutte le relazioni, la chiave primaria è quindi quella dell'entità di Conto. L'attributo che lo caratterizza è Tasso di interesse che indica il valore, concordato tra cliente e banca, di quanto rende mensilmente il deposito su quel conto.

- L'entità ***PRESTITO*** costituisce il servizio creditizio della banca. Essa è caratterizzata innanzitutto da un codice univoco che funge da chiave primaria, garantendo l'identificazione sicura di ogni singolo prestito all'interno del sistema. L'attributo *ammontare* fornisce invece l'informazione relativa alla somma di denaro effettivamente erogata, mentre l'attributo *inizio* registra la data in cui il prestito ha avuto origine. Un aspetto interessante di questa entità è la presenza di un attributo derivato,

somma rate calcolato sulla base dell'insieme delle rate associate a quel prestito. Questo calcolo deriva appunto dalla relazione con l'entità *RATA*, che verrà descritta successivamente.

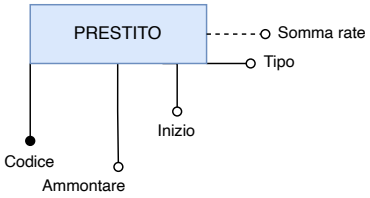

Figura 6: *Entità PRESTITO*

- L'entità ***RATA*** è una entità debole ed ha il compito di rappresentare in modo dettagliato ogni singolo pagamento periodico associato a un determinato prestito. L'identificazione univoca di ciascuna rata è garantita da una chiave primaria composta, costituita dal suo numero (indicante la "posizione" della rata nella sequenza dei pagamenti) e dalla chiave esterna che fa riferimento all'entità Prestito. Tra gli attributi figurano inoltre la *data scadenza*, ossia il giorno entro cui la rata deve essere corrisposta, e la *data pagamento*, che riporta il momento in cui il versamento è stato effettivamente effettuato. Infine, l'attributo *ammontare* specifica l'importo dovuto per quella singola rata.

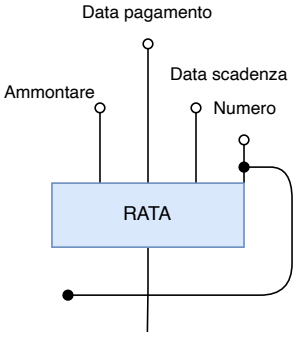

Figura 7: *Entità RATA*

- La relazione ***È CAPO*** collega l'entità *CAPO* con l'entità *FILIALE*, definendo il legame tra il capo di una filiale e la filiale stessa. La cardinalità di (1,1) tra la relazione e l'entità Filiale indica che ogni *FILIALE* ha un solo capo, mentre la cardinalità di (0,1) tra la relazione e l'entità *CAPO* indica che un dipendente può essere al più capo di una sola filiale.

Figura 8: *Relazione È CAPO*

- La relazione ***LAVORA*** collega l'entità *DIPENDENTE* con l'entità *FILIALE*. La cardinalità di (1,1) tra la relazione e l'entità Dipendente indica che ogni dipendente lavora in una e in una sola filiale, mentre la cardinalità di (1,N) tra la relazione e l'entità *FILIALE* indica che in una filiale lavora uno o più dipendenti.

Figura 9: *Relazione LAVORA*

- La relazione ***DI*** lega l'entità *DIPENDENTE* con l'entità *CAPO*. La cardinalità di (1,N) tra la relazione e l'entità *CAPO* indica che un capo dirige uno o più dipendenti, mentre la cardinalità di (1,1) tra la relazione e l'entità *DIPENDENTE* indica che un dipendente ha uno e un solo capo.

Figura 10: *Relazione DI*

- La relazione ***É COMPOSTO*** collega l'entità *PRESTITO* con l'entità *RATA*, dando forma al legame logico tra un finanziamento e i singoli pagamenti previsti per il suo rimborso. Dal lato di *RATA*, la cardinalità è di (1,1), poiché ogni rata è necessariamente associata ad uno e un solo prestito specifico data la natura di *RATA* come entità debole. Dal lato di Prestito, invece, la cardinalità è di (1,N), poiché un singolo prestito può essere suddiviso in una o più rate. In sintesi, questa relazione rispecchia un legame di composizione, dove ogni prestito è scomponibile in un insieme di rate, ma ogni rata non può prescindere dal proprio prestito di appartenenza.

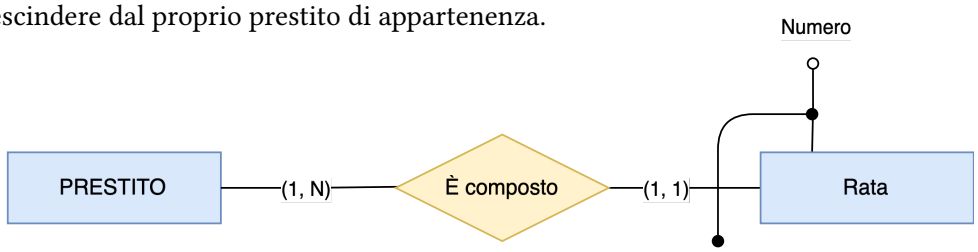

Figura 11: *Relazione È COMPOSTO*

- La relazione ***È ASSOCIATO*** collega l'entità Conto con l'entità Prestito, definendo il legame tra un finanziamento e il conto bancario a cui è associato. Dal lato di *PRESTITO*, la cardinalità è (1,1), poiché ogni prestito deve fare riferimento obbligatoriamente a un solo conto bancario. Dal lato di Conto, invece, la cardinalità è (0,N): questo riflette il fatto che un conto può non avere alcun prestito associato, ma può anche essere collegato a uno o più prestiti contemporaneamente.

Figura 12: *Relazione È ASSOCIATO*

- La relazione ***POSSIEDE*** collega le entità *CLIENTE* e *CONTO*. Un cliente deve possedere almeno un conto e più clienti possono possedere lo stesso conto (caso di conto cointestato), da cui deriva la cardinalità (1, N) della relazione sul lato di *CLIENTE*. D'altro canto un *CONTO* deve essere posseduto da almeno un cliente e più conti possono fare riferimento allo stesso cliente (caso in cui uno stesso cliente ha aperto più conti con la banca), da cui deriva la cardinalità (1, N) della relazione sul lato di *CONTO*. Gli attributi *Operazione* e *Data* sulla relazione indicano l'ultima operazione svolta e la data in cui è stata effettuata. Nel caso di operazione congiunta di più clienti possessori dello stesso conto gli attributi *Operazione/Data* vengono aggiornati per entrambi.

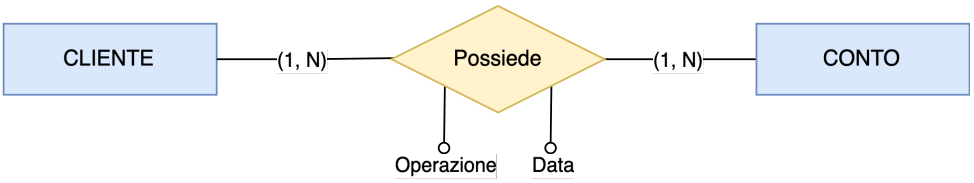

Figura 13: *Relazione POSSIEDE*

- La relazione ***GESTISCE*** lega *DIPENDENTE* e *CLIENTE*. Un sottoinsieme dei dipendente possono seguire le pratiche di un certo numero di clienti della banca, da cui ne deriva la cardinalità (0, N) della relazione sul lato di *DIPENDENTE*. D'altro canto un *CLIENTE* può avere al più un solo gestore che monitora e consiglia le sue attività nella banca, da cui ne deriva la cardinalità (0, 1) della relazione sul lato di cliente.

Figura 14: *Relazione GESTISCE*

- La relazione ***CONTIENE*** collega *FILIALE* a *CONTO* in quanto ogni *CONTO* deve fare riferimento ad una e una sola *FILIALE*. Una filiale può contenere uno o più conti (anche zero se la filiale è appena stata aperta), da cui ne deriva la cardinalità (0, N) della relazione sul lato di FIliale. D'altro canto un *CONTO* deve essere associato ad una e una sola *FILIALE*, da cui ne deriva la cardinalità (1, 1) della relazione sul lato di Conto.

Figura 15: *Relazione CONTIENE*

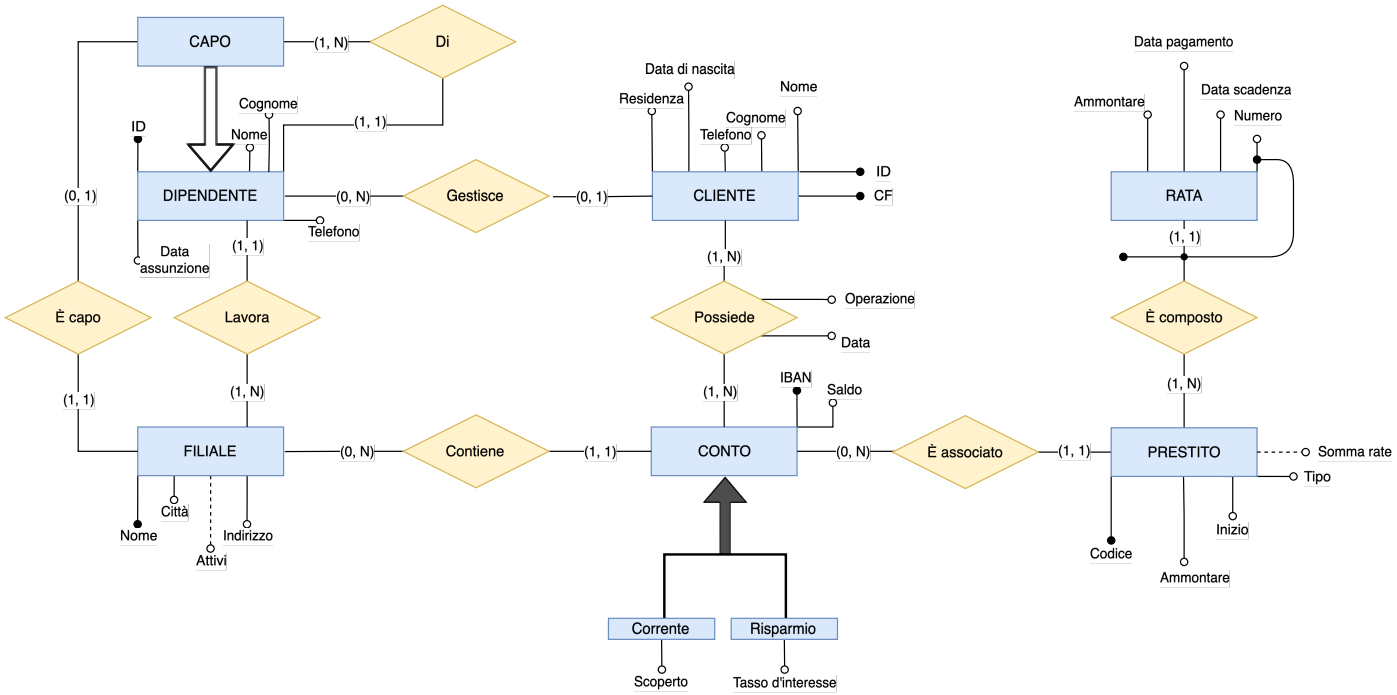

Figura 16: *Schema concettuale nel modello Entità Relazioni*

2.1.1.

2.1.2.

2.2. Schema Concettuale

3. Progettazione Logica

3.1. Tabella dei volumi

Analizziamo ora i passi che abbiamo effettuato per le ridondanze ed i volumi del nostro schema. Nella ottimizzazione delle prestazione e nella semplificazione dello schema ER concettuale verso lo schema ristrutturato abbiamo considerato i volumi dei dati, che sono stati ipotizzati secondo i volumi di una banca reale di riferimento (Banca Intesa San Paolo) e delle operazioni che in seguito elencheremo.

Nome	Costrutto	Volume
Dipendente	Entità	100.000
Cliente	Entità	15.000.000
Filiale	Entità	3.000
Conto	Entità	12.000.000
Conto Corrente	Entità	10.000.000
Conto di Risparmio	Entità	2.000.000
Prestito	Entità	7.000.000
Gestisce	Relazione	10.000.000
Lavora	Relazione	100.000
È capo	Relazione	3.000
È composto	Relazione	7.000.000
Di	Relazione	100.000
Possiede	Relazione	19.000.000
Contiene	Relazione	12.000.000
È associato	Relazione	7.000.000

Tabella 2: *Tabella dei volumi*

3.2. Analisi delle ridondanze

Il primo blocco di operazioni coinvolge l'attributo derivato attivi che produce una ridondanza ed è derivalbile da altre entità, nel nostro caso da Conto, Prestito e Rata, seguendo il caso di attributo derivabile da altre entità secondo funzioni aggregative. Dobbiamo quindi ipotizzare delle operazioni e le loro relative frequenze che vanno a coinvolgere questo attributo ed osservare se è conveniente eliminarlo o mantenerlo. Consideriamo una serie di operazioni e la loro frequenza:

1. interrogazione per leggere il valore attivi di ogni filiale con frequenza 1 volta al giorno,
2. inserimento di un conto nella base di dati con frequenza 150 volte al giorno,
3. inserimento di una operazione in possiede con frequenza 1.000.000 al giorno,
4. aggiornamento di tutti i prestiti con frequenza di 1 volta al mese.

Queste operazioni con la presenza dell'attributo ridondante *attivi* portano ai seguenti costi:

3.2.1. Con ridondanza

Nome	Costrutto	Acecssi	Tipo
Filiale	Entità	3000	Lettura

Tabella 3: *Operazione 1*

Nome	Costrutto	Accessi	Tipo
Conto	Entità	150	Scrittura
Contiene	Entità	150	Scrittura
Possiede	Relazione	150	Scrittura
Filiale	Entità	150	Scrittura
Filiale	Entità	150	Lettura

Tabella 4: *Operazione 2*

Nome	Costrutto	Accessi	Tipo
Possiede	Relazione	1.000.000	Scrittura
Possiede	Relazione	1.000.000	Lettura
Conto	Entità	1.000.000	Scrittura
Conto	Entità	1.000.000	Lettura
Filiale	Entità	1.000.000	Scrittura
Filiale	Entità	1.000.000	Lettura
Contiene	Relazione	1.000.000	Lettura

Tabella 5: *Operazione 3*

Nome	Costrutto	Accessi	Tipo
Rata	Entità		Scrittura
È composto	Relazione		Scrittura
È composto	Relazione		Lettura
Prestito	Entità		Lettura
È associato	Relazione		Lettura
Conto	Entità		Lettura
Contiene	Relazione		Lettura
Filiale	Entità		Lettura
Filiale	Entità		Scrittura

Tabella 6: *Operazione 4*

3.2.2. Senza ridondanza

Nome	Costrutto	Accessi	Tipo
Rata	Entità		Scrittura
È composto	Relazione		Scrittura
È composto	Relazione		Lettura
Prestito	Entità		Lettura
È associato	Relazione		Lettura
Conto	Entità		Lettura
Contiene	Relazione		Lettura
Filiale	Entità		Lettura
Filiale	Entità		Scrittura

Tabella 7: *Operazione 1*

op1: $(1 \text{ lettura}) \cdot 3.000$
op2: $(4 \text{ scrittura}\{\text{conto, contiene, possiede, filiale}\} + 1 \text{ lettura}\{\text{filiale}\}) \cdot 150$ op3: (3 scrittura{Possiede, conto, filiale} + 4 letture{Possiede, conto, filiale, contiene}) x 1.000.000 op4: (3 scritture{Rata, Prestito, Filiale} + 5 Letture{}) x 7.000.000 x 1/30 totale: 12.107.500

Senza ridondanza op1: ((2 letture **4000 {Contenuto, Conto}) + (2 letture** 2333{è associato, Prestito})) **3000 op2: (3 scritture{Possiede, Conto, Contenuto})** 150 op3 : (2 letture{Possiede, Conto}, 2 scritture{Possiede, Conto}) **1.000.000 op4: (1 scrittura, 1 lettura)** 7.000.000 x 1/30 totale: 44.700.900

La seguente analisi ci suggerisce che la conservazione dell'attributo derivato attivi sia utile e quindi lo manteniamo nel nostro schema ristrutturato.

Il secondo blocco di operazioni riguardano la ridondanza introdotta dall'attributo derivato somma rate dell'entità Prestito, anche in questo caso è il caso di un attributo derivato secondo funzioni aggregative e le entità che sono coinvolte sono Rata e Prestito. Possiamo considerare due operazioni che sono: 1- inserimento di una rata una volta al mese per ogni prestito della banca 2- lettura del valore della somma delle rate pagate per ogni prestito con frequenza di 2 volte all'anno. Per questa analisi abbiamo dovuto introdurre una ulteriore ipotesi e cioè il numero medio di rate presenti nella nostra base di dati per ogni prestito. Abbiamo supposto essere questo numero 12, che equivale ad un anno di rate pagate.

Con ridondanza Inserimento di una rata: 2 (scritture) **7.000.000 (prestiti)** 1 (volta al mese) = 28 mln Lettura: 1 (lettura) **7.000.000 (prestiti)** 1/6 mese = 7/6 mln totale: 29.160.000

Senza ridondanza Inserimento di una rata: 1 (scrittura) **7.000.000 (prestiti)** 1 (volta al mese) = 14 mln Lettura = 12 **7.000.000** 1/6 = 84 mln totale: 28.000.000

Per questa ridondanza abbiamo concluso quindi che l'attributo somma rate potesse essere rimosso dal nostro schema e non utilizzato nello schema ER ristrutturato.

3.2.3.

3.2.4.

3.3. Selelezione delle chiavi primarie

3.4. Schema E-R ristrutturato

3.5. Schema Logico

4. Popolamento del database

Per creare il database richiesto, popolarlo e testare le query assegnate abbiamo dovuto seguire una particolare logica affinché tutto venisse inserito correttamente. Infatti si potevano presentare delle problematiche relative a chiavi esterne e/o a dei trigger, ma vediamo nel dettaglio l'ordine delle operazioni che sono state eseguite.

Per prima cosa è stato creato il database, assegnando i volumi dei dati con valori proporzionati alla tabella dei volumi precedentemente proposta. Vengono poi create tutte le tabelle in un ordine preciso; in particolare i vincoli di chiave esterna sono stati aggiunti solo quando tutte le tabelle coinvolte erano esistenti, altrimenti si sarebbe generato un errore.

Vengono poi caricati nel sistema tutti i trigger utilizzati e temporaneamente disabilitati per possibili inconsistenze momentanee nell'inserimento dei dati. La modalità di generazione casuale dei dati è stata pensata in modo tale che, al termine degli inserimenti iniziali, tutto sia coerente e non ci siano errori.

Le prime tabelle popolate sono *FILIALE* e *DIPENDENTE*. Al termine del popolamento vengono eseguiti forzatamente due trigger in maniera tale da assegnare automaticamente i manager (che non erano stai inseriti) e verificare la presenza di eventuali errori (di base i dati sono stati generati consistentemente).

Estratti dai possibili gestori vengono inseriti i clienti, successivamente la tabella *CONTO* con le relative *CONTO CORRENTE* e *CONTO DI RISPARMIO*. Una volta inseriti questi dati è possibile procedere al popolamento della tabella *POSSIEDE* che gestisce tutte le connessioni tra i clienti e i loro conti.

Per la macrocategoria dei prestiti, una volta generati quest'ultimi e le relative rate andiamo, tramite apposito script, a pagare le rate che hanno una data di scadenza antecedente a quella odierna. Inseriti

tutti i prestiti aggiorniamo l'attributo *attivi* della tabella *FILIALE* in maniera automatica sui dati inseriti e al termine riattiviamo tutti i trigger.

Gli script utilizzati non potevano essere sempre sostituiti dai trigger, infatti non era possibile tenerli tutti attivi e inserire tutti i valori in maniera ordinata e raggruppati per tabelle, ma avremmo dovuto fare attenzione volta per volta. Degli esempi di inserimenti di record sono presentati più avanti.

4.1. Test

Finito di popolare tutto il database ci assicuriamo tramite dei test che tutto sia perfettamente funzionante, che rispetti i requisiti che ci siamo imposti e che ci dia i risultati attesi. Questa verifica viene effettuata confrontando il risultato ottenuto dalle operazioni con i risultati attesi.

4.2. Test su relazione Dipendente-Filiale

1. Tentiamo di modificare la filiale di riferimento di un manager senza togliergli il ruolo nell'altra filiale. Il trigger ci protegge e ci vieta l'inserimento (un dipendente non può lavorare nella filiale A ed essere manager della filiale B).
2. Simile al precedente, proviamo ad assegnare il ruolo di manager di una filiale a un dipendente che lavora presso una filiale diversa. Il trigger blocca l'azione e ci restituisce l'errore (la modifica non viene effettuata).
3. Inseriamo un nuovo dipendente: non è necessario specificare il campo manager in quanto il trigger apposito si occupa di ricercare l'id del manager nella filiale dove lavora il nuovo dipendente e assegnare il campo corrispondente.
4. Come il caso (3) ma con l'aggiunta che questo dipendente diventi manager della filiale in cui lavora. Il trigger che viene innescato sulla modifica del campo manager (che passa da −1 [non manager] a un id di filiale valido) provvede ad aggiornare il campo manager di tutti i dipendenti che lavorano nella filiale dove è appena stato modificato il manager.
5. Controlliamo una semplice operazione di rimozione di un dipendente che non è manager.

4.3. Test su relazione Prestito-Rata

1. Inseriamo un nuovo prestito. Le rate relative verranno generate in maniera automatica dal trigger che si occupa di andare a recuperare il valore di "mensilità" e generare altrettanti record nella tabella "Rate" riempiendo in maniera adeguata tutti i campi.
2. Modifichiamo la data di pagamento di una data, portandola da NULL a una data valida. Il controllo del trigger sarà di verificare che non ci siano rate precedenti ancora da pagare.

4.4. Test su relazione Conto-Filiale

1. Simuliamo un versamento e un prelievo, quindi andiamo a modificare il valore del saldo dei conti. A questo punto dei trigger controllano (solo nel secondo caso) che il prelievo possa essere effettuato, quindi che il saldo un numero valido (non minore dello scoperto), dopodiché in entrambi i casi vengono automaticamente aggiornati gli attivi delle filiali. Lo scopo del test è comunque di verificare che il saldo venga correttamente modificato
2. Controlliamo che il trigger dei saldi non validi funzioni, forzando la modifica di un saldo a un valore non valido. Ci attendiamo un errore.
3. Simile al primo test con il focus sull'aggiornamento degli attivi della filiale di riferimento.
4. Proviamo a inserire un iban valido nella tabella "Conto" (necessario per i vincoli di chiave esterna) e poi nella tabella "Conto Corrente". Questo non dovrebbe generare problemi. Proviamo a inserire

l'iban anche in "Conto di Risparmio", il trigger dovrebbe vietare tale operazione e, dato che siamo all'interno di una transazione, tutti e tre gli inserimenti vengono rimossi (rollback).

5. Query

Dopo aver verificato che anche i test restituivano i risultati attesi, procediamo con l'esecuzione delle query:

5.1. QUERY 1:

Restituire il numero medio di rate dei prestiti associati a conti nelle filiali di Udine.

Richiesta immediata, necessario l'utilizzo della funzione `AVG()`

5.2. QUERY 2:

Restituire i clienti con solo conti di risparmio in filiali che hanno tra i 30 e i 32 dipendenti.

Per comodità è stata creata una vista dove veniva fatta una restrizione della tabella delle filiali, tenendo solamente quelle che rispettavano i vincolo sui clienti. La query poi si appoggia su questa vista per cercare i clienti che hanno almeno un conto di risparmio in queste filiali e che non hanno nessun conto corrente associato.

5.3. QUERY 3:

Restituire i capi che gestiscono almeno 3 clienti che possiedono almeno 100 000€.

La vista creata è una restrizione sui clienti che rispettano il vincolo. È stata effettuata con l'utilizzo della funzione SUM() poiché il saldo era relativo a tutti i conti posseduti. Per validare un capo è stato fatto il prodotto cartesiano triplo della tabella generata dalla vista precedente e dopo essere stati selezionati solamente le righe con gestore uguale, è stato controllato che i clienti fossero tutti e tre diversi.

5.4. QUERY 4:

Restituire i dipendenti non capi che gestiscono esattamente 2 clienti, uno con solo conti correnti e uno son solo conti di risparmio.

La prima (seconda) vista seleziona solamente i clienti che hanno almeno un conto corrente (di risparmio) e che non hanno nessun conto di risparmio (corrente). La query innanzitutto seleziona i dipendenti non capo (con la verifica id <> capo) e poi controlla che esista un unico cliente nella prima vista e un unico cliente nella seconda vista.

5.5. QUERY 5:

Restituire il cliente con il prestito più alto nella filiale di Roma che non ha come gestore un dipendente con meno di 3 anni di esperienza. La prima vista ci restringe i possibili clienti a quelli che hanno un gestore assunto da almeno 3 anni.

La seconda vista, a partire dalla prima, fa un ulteriore filtro prendendo i clienti solo della filiale di Roma. La query si occupa di verificare, per ogni cliente, che tra i clienti della seconda vista non ce ne sia qualcuno con saldo maggiore del proprio, in tal caso stampa il cliente.

(test)

sql

Creazione Tabelle

```sql
CREATE SCHEMA banca
    AUTHORIZATION enrperes;

COMMENT ON SCHEMA banca
     IS 'Il database per la gestione delle filiali di una banca, progetto di Basi di Dati.';

SET search_path TO banca;

-- Creazione delle tabelle del database
-- Tabella dipendente senza FOREIGN KEY
CREATE TABLE dipendente (
    id SERIAL PRIMARY KEY,
    nome VARCHAR(30),
    cognome VARCHAR(30),
    data_assunzione DATE NOT NULL,
    telefono VARCHAR(15) CHECK (telefono ~ '^\+?[0-9]+$') UNIQUE,
    filiale VARCHAR(30) NOT NULL,
    capo INT
);
```