

Corso di Basi di Dati A.A. 2020/21
Progetto "Fly Away"

Federico Matteoni
Mat. 530257

08/01/2021

1 Descrizione del dominio

Il sistema si concentra sulla gestione dei biglietti, della loro prenotazione e vendita, sulla gestione delle promozioni e sulla gestione degli utenti.

Un utente può essere un agente di viaggio o un cliente. Un cliente può essere una società, con nome e partita iva, o un privato, con i dati anagrafici e il codice fiscale.

Un cliente può abbonarsi al programma fedeltà della compagnia, diventando infedele se non acquista biglietti per più di due anni.

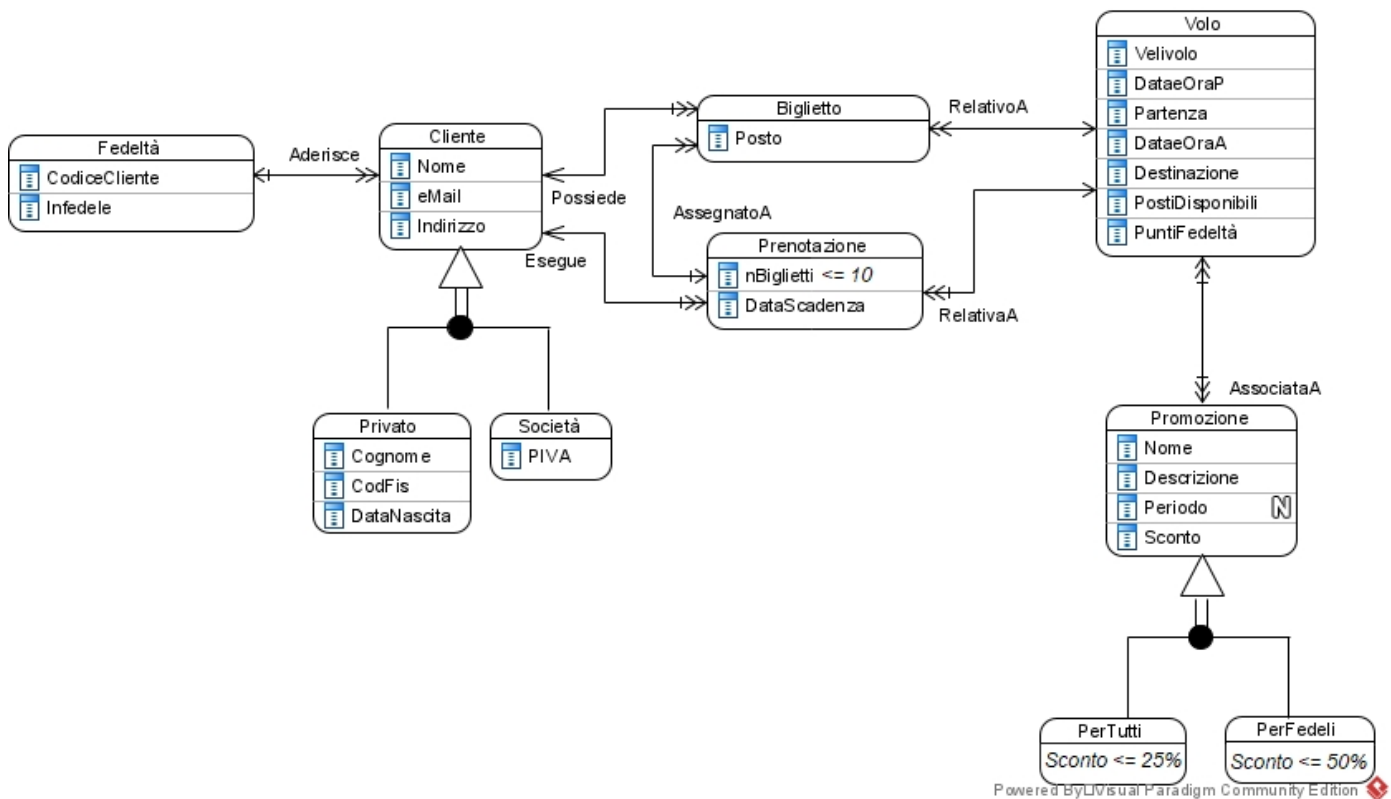
Una prenotazione permette al cliente di prenotare non più di dieci biglietti su un singolo volo. Un cliente può avere più di una prenotazione attiva ma solo se tutte su voli diversi, e la prenotazione scade 3 giorni prima del volo.

Un biglietto può essere legato ad una prenotazione o acquistato direttamente, e porta le indicazioni del volo e del cliente.

Un volo, con dati e orari di partenza e arrivo e informazioni sui posti disponibili, può avere una o più promozioni ad esso legate. Ogni volo assegna un quantitativo di punti agli utenti fedeli.

Una promozione può essere legata ad un periodo di tempo o a determinati voli. Una prenotazione può essere aperta a tutti i clienti o essere esclusiva del programma Fedeltà, prevedendo uno sconto massimo del 25% nel primo caso e del 50% per i fedeli.

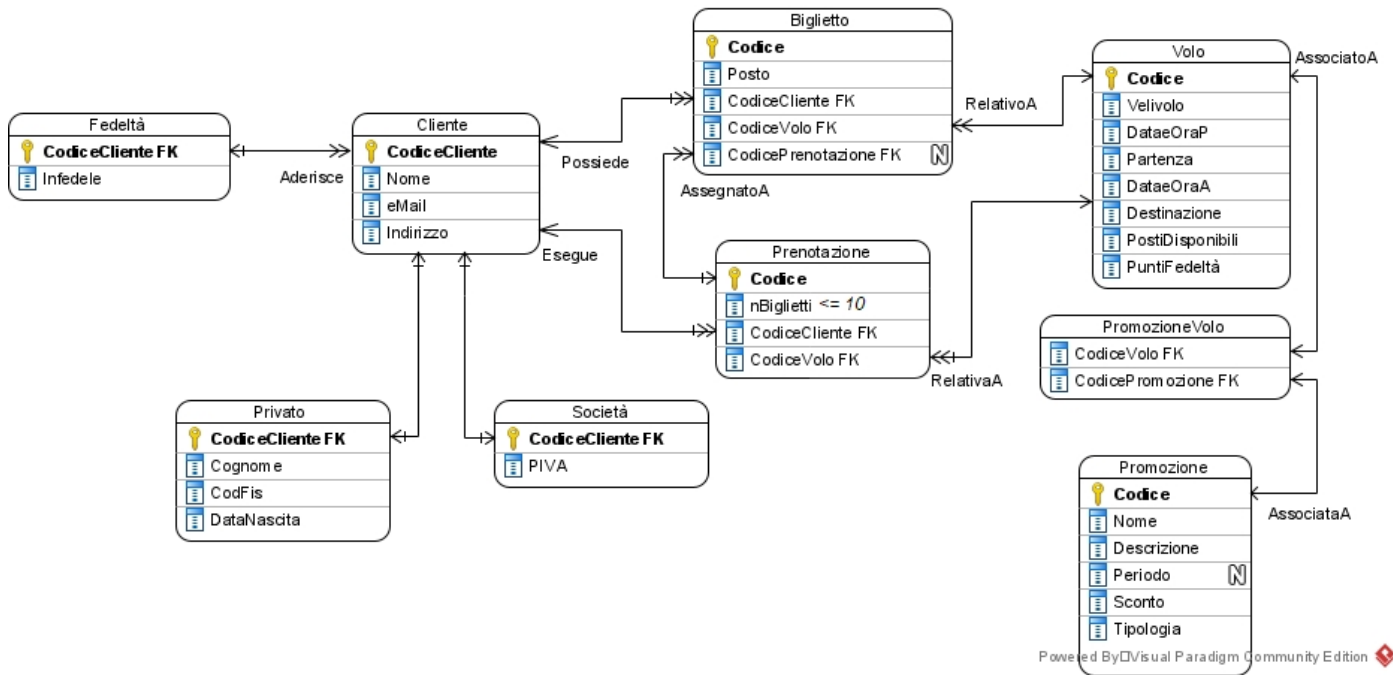
2 Schema Concettuale



Il numero di righe nella tabella Biglietto legate alla medesima prenotazione non può essere superiore a 10.

Ogni prenotazione assegnata al solito cliente deve essere legata ad un volo diverso.

3 Schema Logico Relazionale



Fedeltà(CodiceCliente*, Infedele)

Cliente(CodiceCliente, Nome, eMail, Indirizzo)

Privato(CodiceCliente*, Cognome, CodFis, DataNascita)

Società(CodiceCliente*, PIVA)

Biglietto(Codice, Posto, CodiceCliente*, CodiceVolo*, CodicePrenotazione*)

Prenotazione(Codice, nBiglietti, CodiceCliente*, CodiceVolo*)

Volo(Codice, Velivolo, DataeOraP, Partenza, DataeOraA, Destinazione, PostiDisponibili, PuntiFedeltà)

PromozioneVolo(CodiceVolo*, CodicePromozione*)

Promozione(Codice, Nome, Descrizione, Periodo, Sconto, Tipologia)

Tutte le dipendenze funzionali non banali presenti sono dipendenti da una chiave primaria. Un esempio, è la dipendenza $\text{CodiceCliente}^* \rightarrow \text{Cognome}$, nella relazione Privato: il CodiceCliente* determina univocamente il campo Cognome, ma CodiceCliente* è chiave primaria della relazione.

Una dipendenza funzionale che poteva essere presente è la seguente: se si aggiungesse un campo DataScadenza alla relazione Prenotazione, esso sarebbe univocamente determinato dal campo CodiceVolo*, che è chiave primaria della relazione Volo (in quanto la data di scadenza di una prenotazione in una istanza valida di Prenotazione è tre giorni prima del volo relativo), ma non è chiave della relazione Prenotazione.

4 Query SQL

1. Uso di proiezione, join e restrizione

La lista di tutte le società che hanno prenotato almeno un biglietto nell'anno corrente.

```
SELECT Nome, PIVA, eMail
FROM Cliente
JOIN Prenotazione ON Cliente.CodiceCliente = Prenotazione.CodiceCliente
JOIN Volo ON Prenotazione.CodiceVolo = Volo.Codice
JOIN Societa ON Cliente.CodiceCliente = Societa.CodiceCliente
WHERE DataeOraP.YEAR = 2021
```

2. Uso di group by con having, where e sort

La classifica degli utenti privati fedeli con più di N punti su voli negli ultimi 5 anni.

```
SELECT Cognome, Nome, COUNT(PuntiFedelta) as Punti
FROM Cliente
JOIN Biglietto ON Cliente.CodiceCliente = Biglietto.CodiceCliente
JOIN Volo ON Biglietto.CodiceVolo = Volo.Codice
JOIN Fedelta ON Cliente.CodiceCliente = Fedelta.CodiceCliente
JOIN Privato ON Cliente.CodiceCliente = Privato.CodiceCliente
WHERE DataeOraP.YEAR >= 2016
GROUP BY Cliente.CodiceCliente
HAVING Punti > N
ORDER BY Punti
```

3. Uso di join, group by con having e where

La lista delle promozioni sui voli nel 2020 su più di 5 voli, specificando su quanti voli è stata applicata ogni promozione.

```
SELECT Nome, COUNT(CodiceVolo) AS Voli
FROM Promozione
JOIN PromozioneVolo
    ON Promozione.Codice = PromozioneVolo.CodicePromozione
JOIN Volo on PromozioneVolo.CodiceVolo = Volo.Codice
WHERE Volo.DataeOraP.YEAR = 2020
GROUP BY PromozioneVolo.CodicePromozione
HAVING Voli > 5
```

4. Uso di select annidata con quantificazione esistenziale

La lista delle promozioni su voli che assegnano meno di N punti fedeltà.

```
SELECT Nome, Descrizione
FROM Promozione
JOIN PromozioneVolo
    ON Promozione.Codice = PromozioneVolo.CodicePromozione
WHERE EXISTS (SELECT Volo.Codice
               FROM Volo
               WHERE Volo.Codice = PromozioneVolo.CodiceVolo
               AND Volo.PuntiFedelta < N)
```

5. Uso di select annidata con quantificazione universale

La lista delle promozioni su periodo.

```
SELECT Nome, Descrizione
FROM Promozione
WHERE NOT EXISTS (SELECT *
                  FROM PromozioneVolo
                  WHERE Promozione.Codice = PromozioneVolo.CodicePromozione)
```

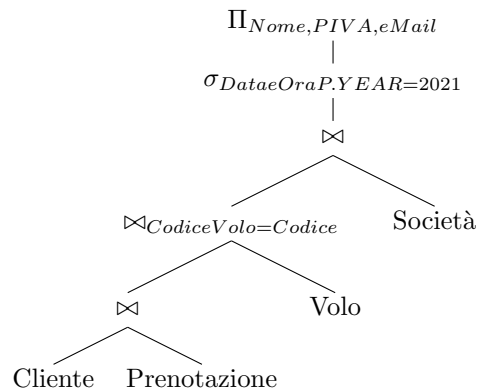
6. Uso di subquery di confronto quantificato usando una subquery
La lista dei voli che forniscono più punti della media

```
SELECT Partenza, Destinazione
FROM Volo
WHERE PuntiFedelta > (SELECT AVG(PuntiFedelta)
FROM Volo)
```

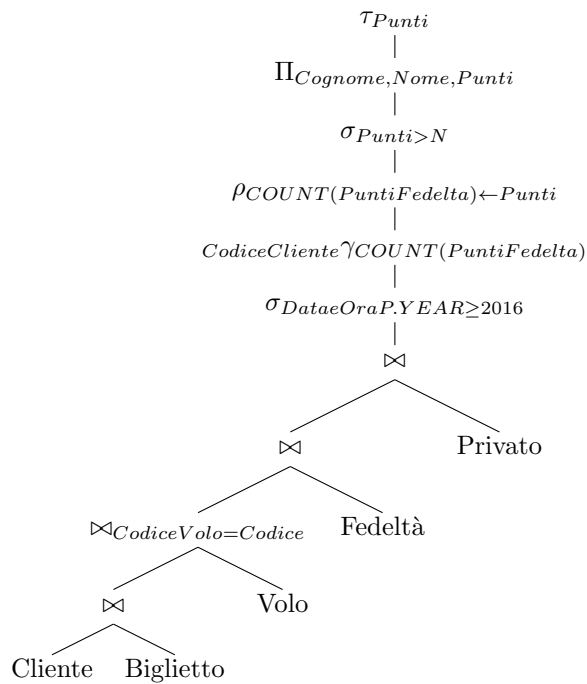
5 Piani di accesso

1. Scrivere un piano di accesso logico delle query 1, 2 e 3

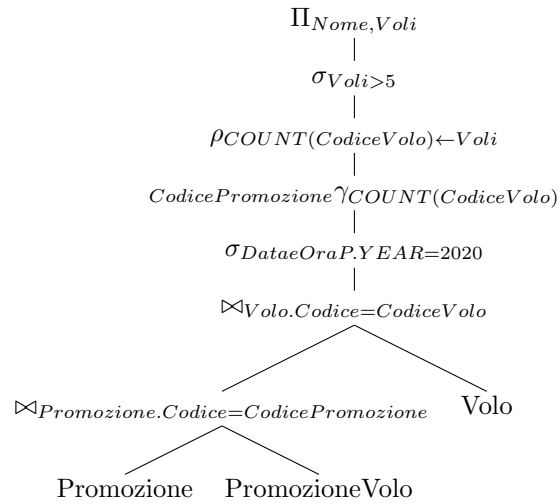
1.



2.

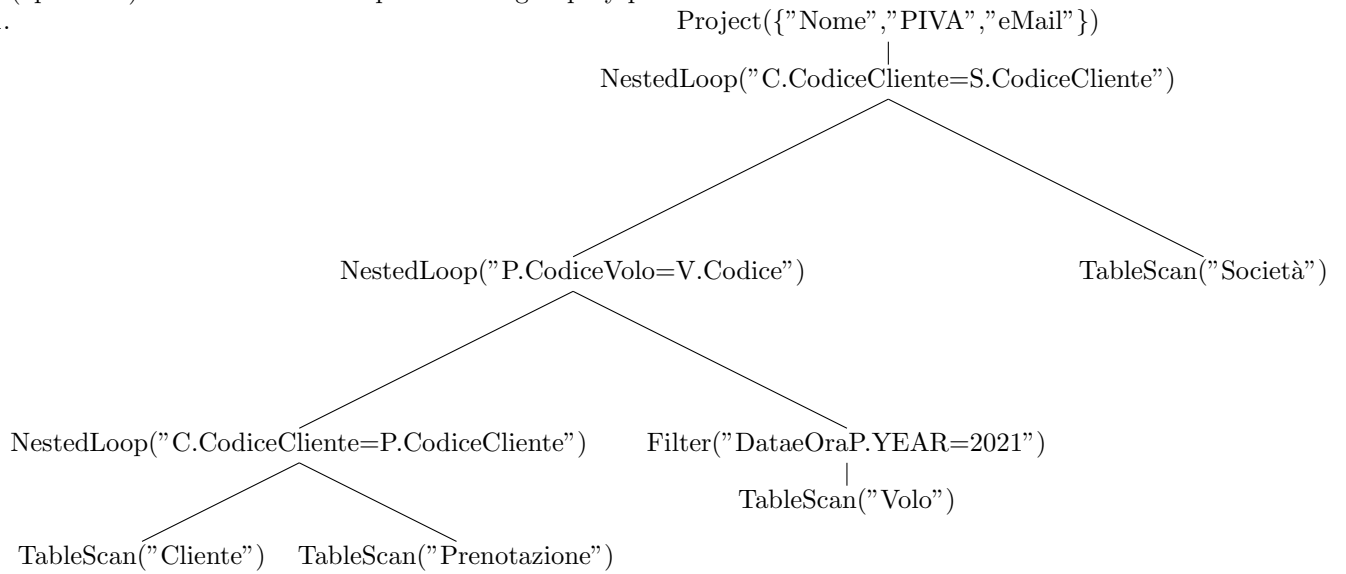


3.



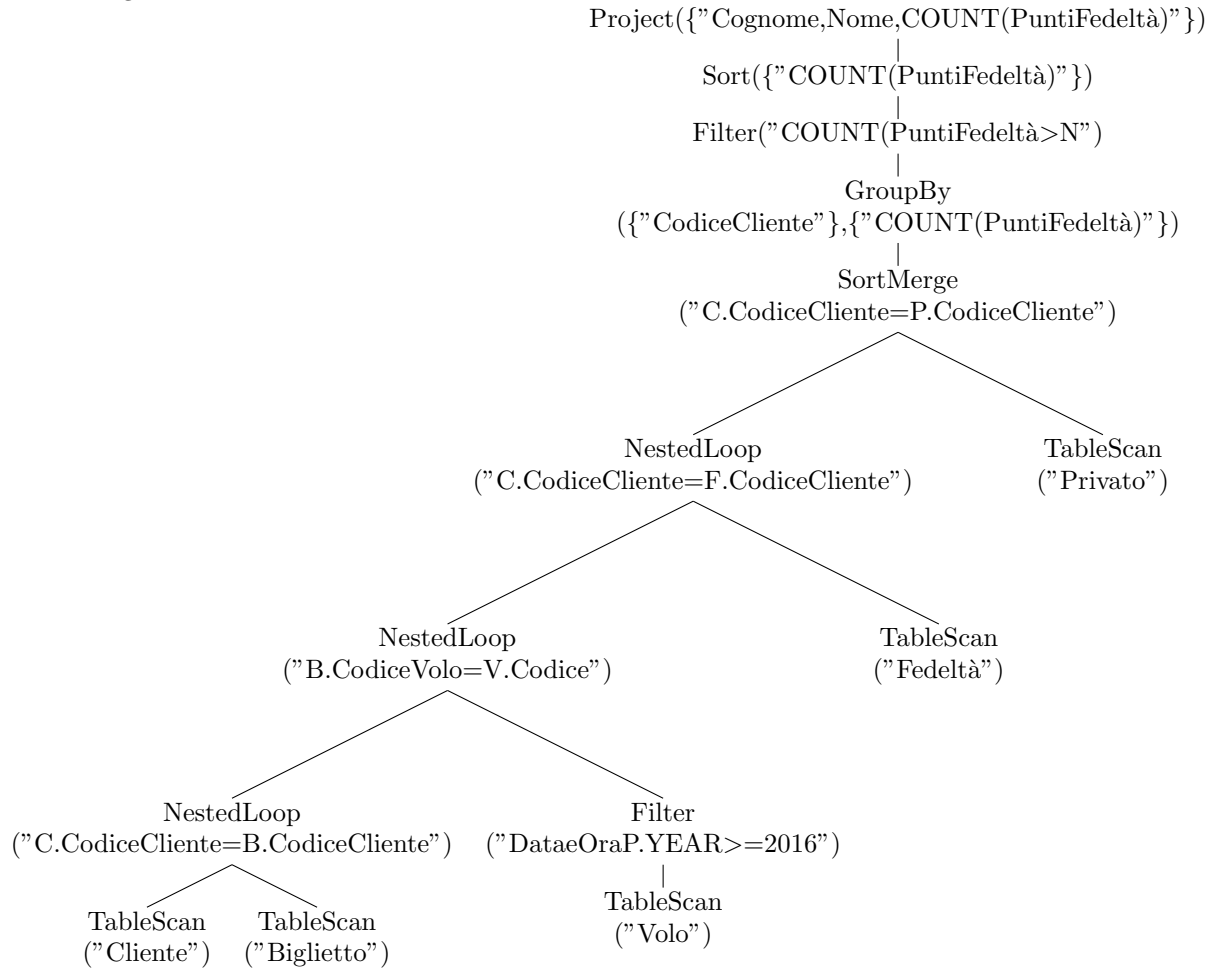
2. Scrivere un piano di accesso fisico efficiente per i tre piani di accesso logico al punto 1 che non fanno uso di indici e (opzionale) verificare se la sort prima della group by può essere evitata

1.

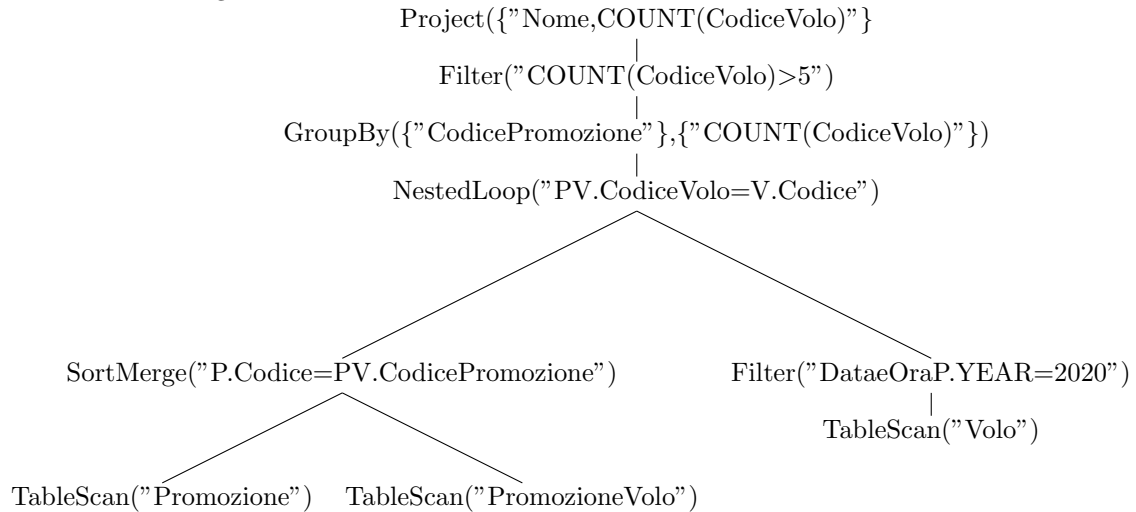


Per questioni di spazio, laddove possibile i nomi delle tabelle sono stati abbreviati con le loro iniziali.

2. In questo caso la Sort prima di GroupBy può essere evitata ordinando le tabelle sugli attributi di giunzione con SortMerge.

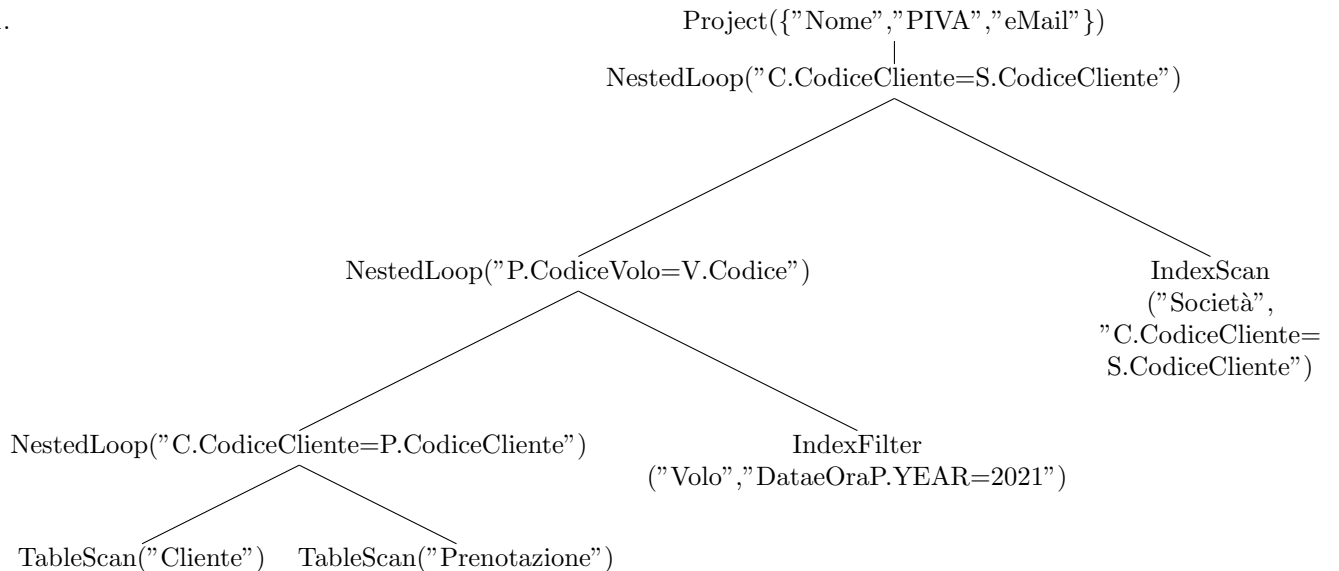


3. Anche in questo caso la Sort prima di GroupBy può essere evitata ordinando le tabelle sugli attributi di giunzione con SortMerge.



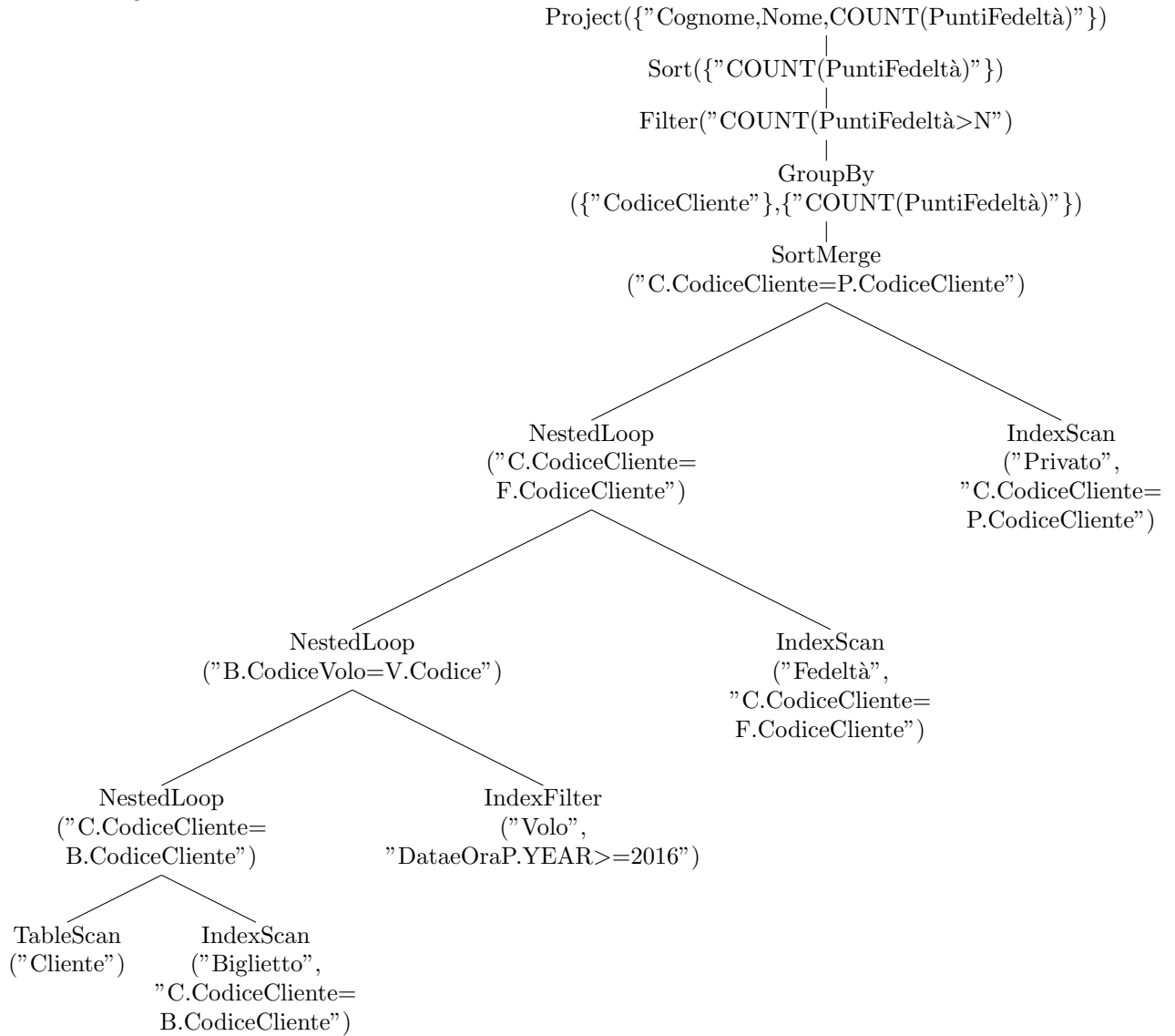
3. Scrivere un piano di accesso fisico efficiente per i tre piani di accesso logico al punto 1 che fanno uso di due indici (o comunque del numero massimo di indici possibili) e (opzionale) verificare se la sort prima della group by può essere evitata

1.



Per questioni di spazio, laddove possibile i nomi delle tabelle sono stati abbreviati con le loro iniziali.

2. In questo caso la Sort prima di GroupBy può essere evitata ordinando le tabelle sugli attributi di giunzione con SortMerge.



3. Anche in questo caso la Sort prima di GroupBy può essere evitata ordinando le tabelle sugli attributi di giunzione con SortMerge.

