

Corso di Basi di Dati A.A. 2019/20  
Progetto BD VIII

Federico Matteoni  
Mat. 530257

24/03/2021

# 1 Descrizione del dominio

Uno studio professionale vuole tenere traccia delle pratiche in corso.

Si vuole tenere traccia dei clienti, che possono essere persone o organizzazioni.

Di ogni cliente, ci interessano il nome, l'indirizzo ed un recapito telefonico.

Delle persone, ci interessano anche il cognome ed il codice fiscale.

Delle organizzazioni ci interessa anche la partita IVA e quali persone svolgono un ruolo al suo interno, ed il ruolo svolto.

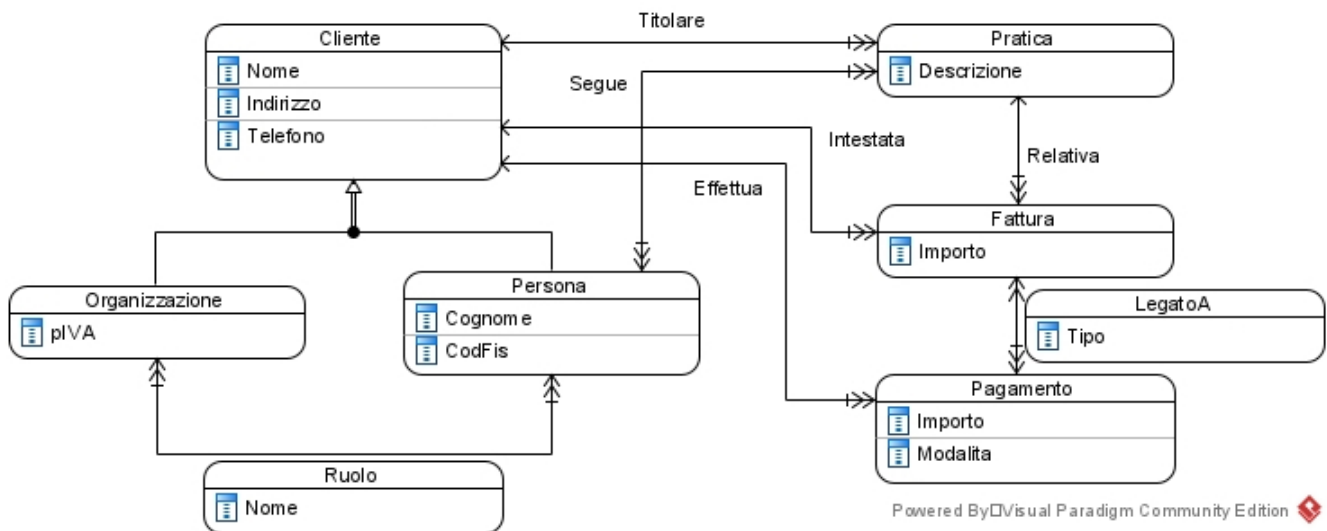
Si vuole tenere traccia delle pratiche, di cui un cliente è titolare.

Se il titolare è un'organizzazione, ci interessa sapere le persone che seguono tale pratica.

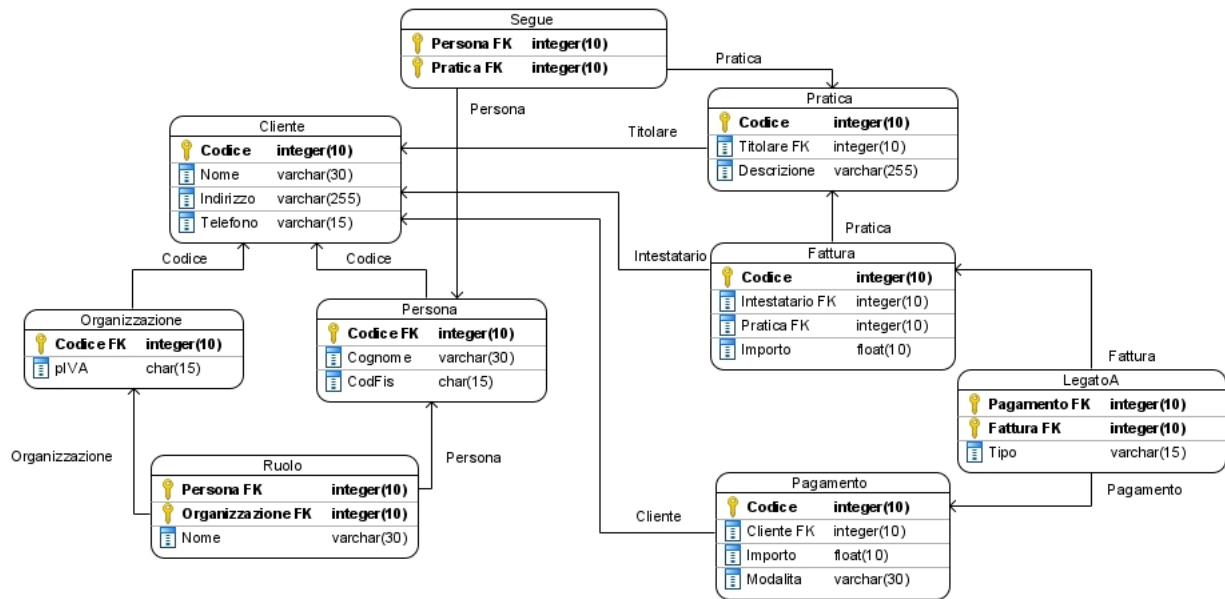
Si emettono fatture intestate ad un cliente e relative ad una pratica.

I clienti effettuano pagamenti, totali o parziali, legati ad una o più fatture e di cui ci interessa sapere la modalità con cui sono stati eseguiti.

## 2 Schema Concettuale



### 3 Schema Logico Relazionale



Powered By DVVisual Paradigm Community Edition

#### Notazione testuale

Cliente(Codice, Nome, Indirizzo, Telefono)

Organizzazione(Codice\*, pIVA)

Persona(Codice\*, Cognome, CodFis)

Ruolo(Persona\*, Organizzazione\*, Nome)

Pratica(Codice, Titolare\*, Descrizione)

Segue(Persona\*, Pratica\*)

Fattura(Codice, Intestatario\*, Pratica\*, Importo)

Pagamento(Codice, Cliente\*, Importo, Modalità)

LegatoA(Pagamento\*, Fattura\*, Tipo)

#### Vincoli non catturati graficamente

## 4 Query SQL

1. Uso di proiezione, join e restrizione

La lista dei clienti con pagamenti superiori a 1000 euro.

```
SELECT Nome  
FROM Cliente JOIN Pagamento ON Cliente.Codice = Pagamento.Codice  
WHERE Importo > 1000
```

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

La classifica delle modalità di pagamento per somma d'importi, per somme maggiori di 1000 euro e singoli pagamenti maggiori di 500 euro.

```
SELECT Modalita, SUM(Importo) AS Totale  
FROM Pagamento  
WHERE Importo > 500  
GROUP BY Modalita  
HAVING Tot > 1000  
ORDER BY Tot
```

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

La lista delle fatture con più di 10 pagamenti inferiori ai 500 euro.

```
SELECT Fattura, COUNT(LegatoA.Pagamento) AS Pagamenti  
FROM LegatoA JOIN Pagamento ON LegatoA.Pagamento = Pagamento.Codice  
WHERE Importo < 500  
GROUP BY LegatoA.Fattura  
HAVING Pagamenti > 10
```

4. Uso di select annidata con quantificazione esistenziale

Le persone che hanno almeno un ruolo in un'organizzazione.

```
SELECT Codice  
FROM Persona  
WHERE EXISTS (SELECT * FROM Ruolo WHERE Ruolo.Persona = Persona.Codice)
```

5. Uso di select annidata con quantificazione universale

Le persone che non seguono alcuna pratica.

```
SELECT Codice  
FROM Persona  
WHERE Codice NOT IN (SELECT Persona FROM Segue)
```

6. Uso di subquery di confronto quantificato usando una subquery

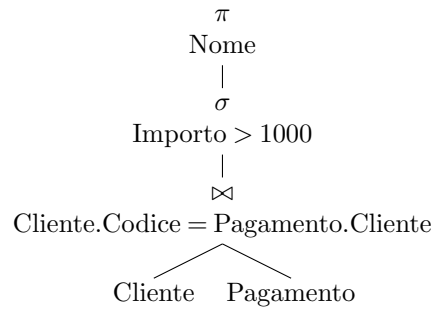
I pagamenti superiori alla media.

```
SELECT *  
FROM Pagamento  
WHERE Importo > (SELECT AVG(Importo) FROM Pagamento)
```

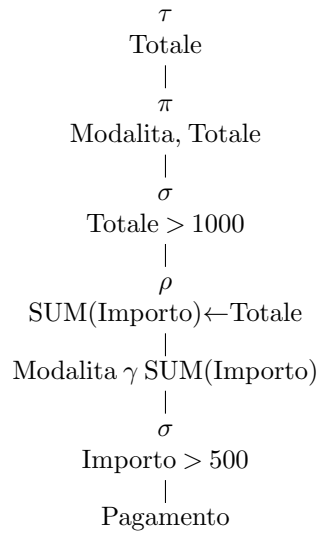
## 5 Piani di accesso

1. Scrivere un piano di accesso logico delle query 1, 2 e 3 abbreviati con le iniziali.

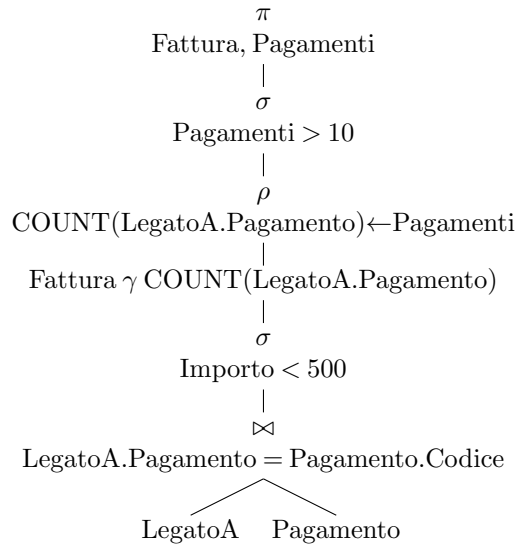
1.



2.

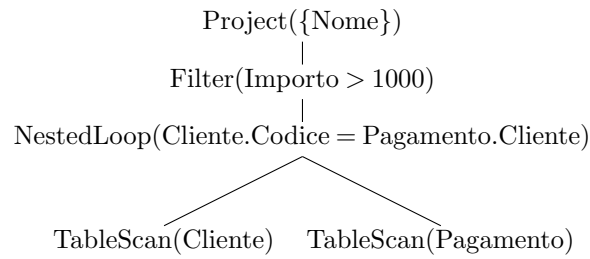


3.

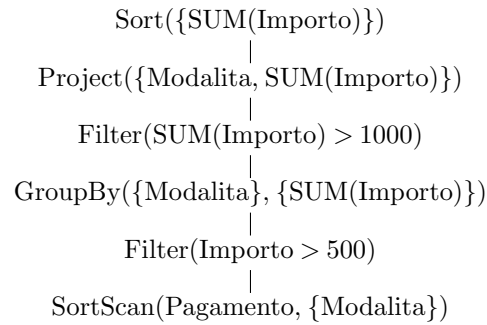


2. Scrivere un piano di accesso fisico efficiente per i tre piani di accesso logico al punto 1 che non fanno uso di indici e (opzionale) verificare se la sort prima della group by può essere evitata.

1.

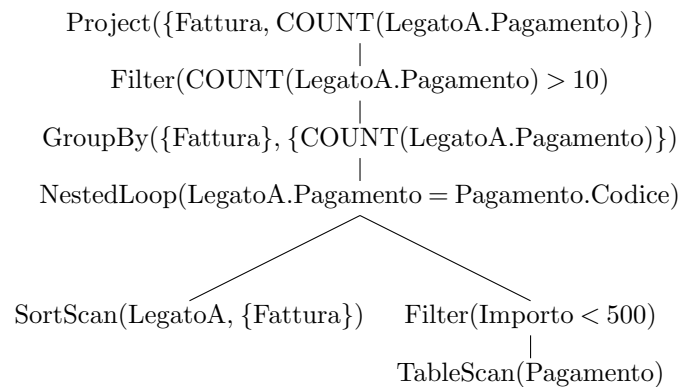


2.



Posso evitare la Sort prima della GroupBy ordinando immediatamente la tabella Pagamento sull'attributo Modalita in fase di scansione.

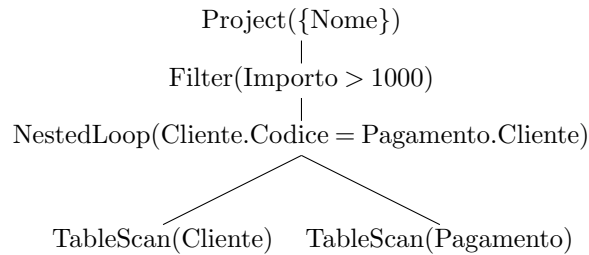
3.



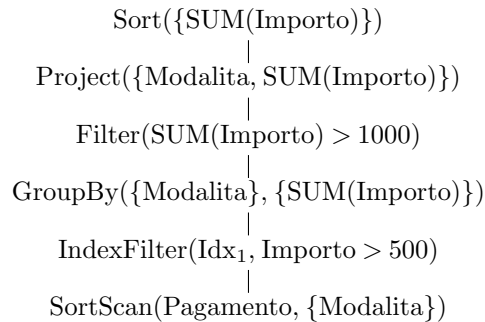
Posso evitare la Sort prima della GroupBy ordinando immediatamente la tabella LegatoA sull'attributo Fattura in fase di scansione, poiché NestedLoop mantiene l'ordinamento della tabella esterna.

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.



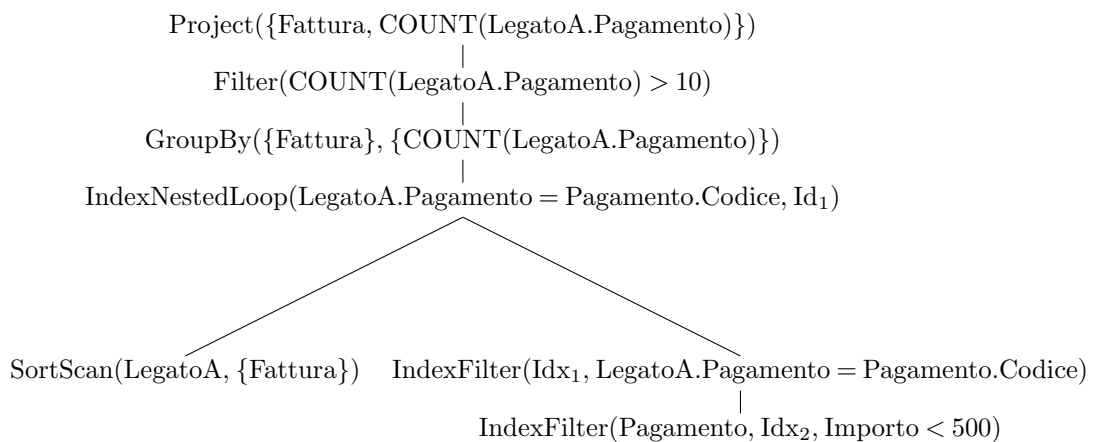
2.



Con  $Idx_1$  un indice su `Pagamento.Importo`

Posso evitare la Sort prima della GroupBy ordinando immediatamente la tabella `Pagamento` sull'attributo `Modalita` in fase di scansione.

3.



Con  $Idx_1$  un indice su `Pagamento.Codice`,  $Idx_2$  un indice su `Pagamento.Importo`

Posso evitare la Sort prima della GroupBy ordinando immediatamente la tabella `LegatoA` sull'attributo `Fattura` in fase di scansione, poiché `NestedLoop` mantiene l'ordinamento della tabella esterna.