



Università di Pisa
Dipartimento di Informatica
Corso di Laurea in Informatica

Corso di Basi di Dati 244AA, prof. Giorgio Ghelli

Progetto di Basi di Dati “Un giorno al Museo”

Lorenzo Pernigoni
Matricola **597958**, Corso A

Consegna: 5 giugno 2021

1 Descrizione del dominio

Il software “Un giorno al Museo” gestisce il sistema museale della regione Toscana.

Ogni **museo** ha un nome, un indirizzo, un proprio sito web e diversi ambienti.

Ogni **ambiente** ha un nome e può essere un ambiente di servizio oppure una sala.

Gli **ambienti di servizio** sono quelli che non contengono opere ed è di interesse sapere un orario approssimativo di apertura e chiusura.

Le **sale** sono stanze che contengono opere.

Ogni sala ha un **tipo**, ovvero può essere una sala museale, per mostre oppure esclusiva. Anche se per ora esistono solamente tre tipi di sala, deve essere lasciata la possibilità di aggiungerne altri in futuro.

Le sale comunicano tra loro attraverso dei **varchi** che hanno un nome. La sala x comunica con la sala y se un varco di x è uguale a un varco di y.

Il sistema tiene traccia delle sale visitate dai visitatori. Di ogni **visita** si vuole conoscere la data e l'ora di ingresso nella sala. Se, nello stesso giorno, un visitatore entra n volte nella stessa sala, vengono registrate n visite.

Ogni **visitatore** deve fornire il nome, il cognome, la data di nascita e, facoltativamente, l'indirizzo e-mail e il numero di cellulare.

L'indirizzo e-mail dei visitatori serve per poter spedire delle **newsletter** bimensili che hanno un titolo e un argomento.

Per poter visitare un museo serve un abbonamento oppure un biglietto.

Gli **abbonamenti** (o abbonamenti fisici) possono essere singoli o multipli. Gli abbonamenti multipli sono associati a diversi visitatori anche se l'acquisto è effettuato da una sola persona. Di ogni abbonamento occorre sapere se è attivo e il tipo.

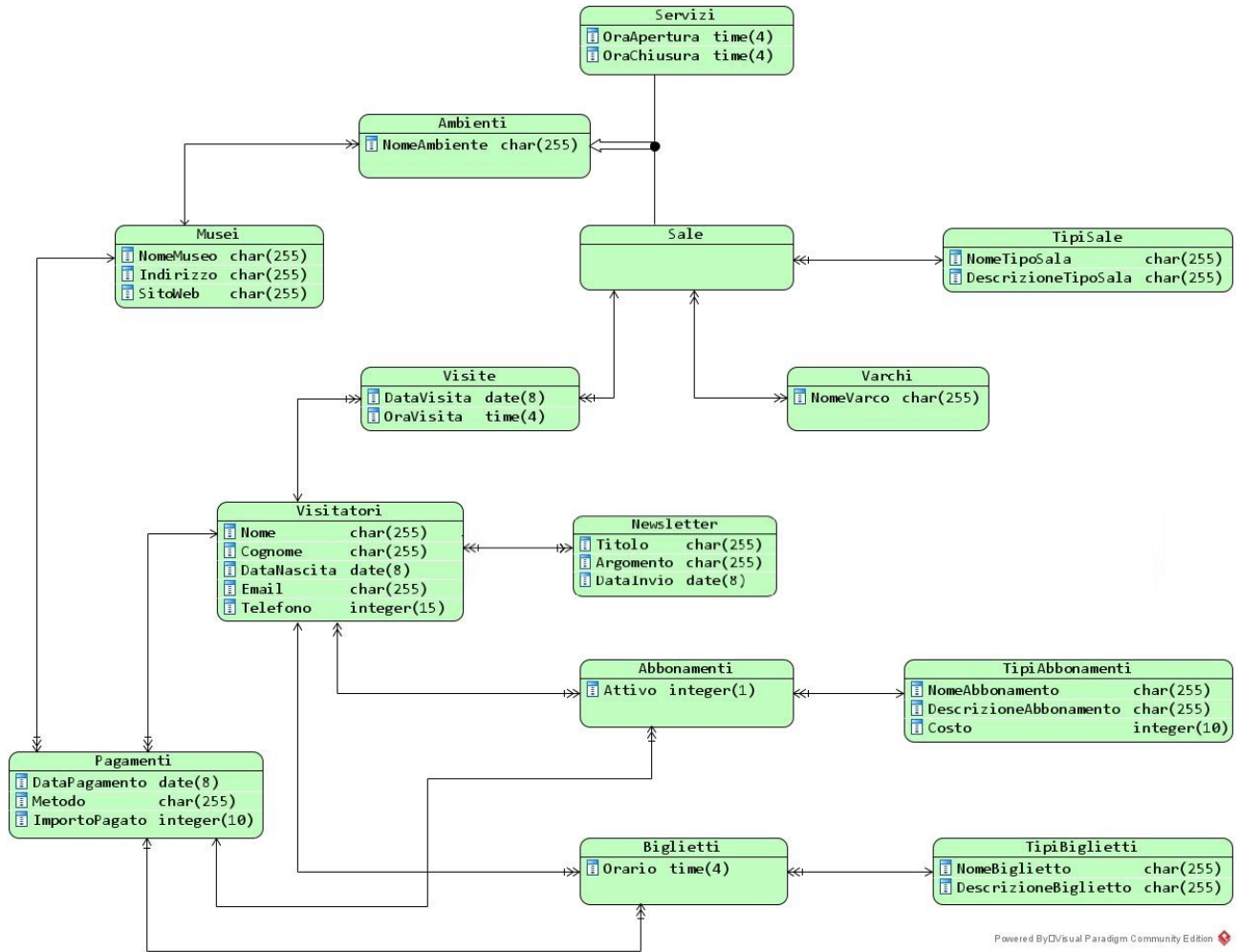
Il **tipo di abbonamento** (o abbonamento astratto) ha un nome, una descrizione e un costo. Per ora esistono solamente abbonamenti standard e speciali, ma deve essere lasciata la possibilità di aggiungere altri tipi di abbonamento in futuro.

I **biglietti** (o biglietti fisici) sono associati ad una sola persona, hanno un orario riferito a quando è stato preso e un tipo.

Il **tipo di biglietto** (o biglietto astratto) ha un nome e una descrizione. Per ora esistono solamente biglietti bianchi e verdi, ma deve essere lasciata la possibilità di aggiungere altri tipi di biglietto in futuro. Il costo non si conosce a priori, dato che dipende dal numero di sale visitate (biglietti bianchi) oppure dalla durata della permanenza nel museo (biglietti verdi).

È necessario tenere traccia dei **pagamenti** di abbonamenti e/o biglietti. Ogni pagamento ha una data, un metodo di pagamento e un importo pagato. Ogni pagamento è associato al visitatore che lo ha effettuato e al museo in cui è avvenuto. Biglietti e/o abbonamenti devono essere pagati nella stessa data in cui sono stati presi.

2 Schema concettuale



2.1 Vincoli

2.1.1 Vincoli intrarelazionali

- I seguenti attributi sono gli unici che possono assumere il valore NULL:
 - *Visitatori.Email*: i visitatori non sono obbligati a fornire l'indirizzo e-mail.
 - *Visitatori.Telefono*: i visitatori non sono obbligati a fornire il numero di cellulare.
- *Abbonamenti.Attivo* è un booleano che vale 1 se l'abbonamento in questione è attivo, 0 se è terminato.
- *Visitatori.Telefono* non può avere meno di 9 cifre.
- *Pagamenti.ImportoPagato* deve essere > 0 .
- *TipiAbbonamenti.Costo* deve essere > 0 .

2.1.2 Vincoli interrelazionali

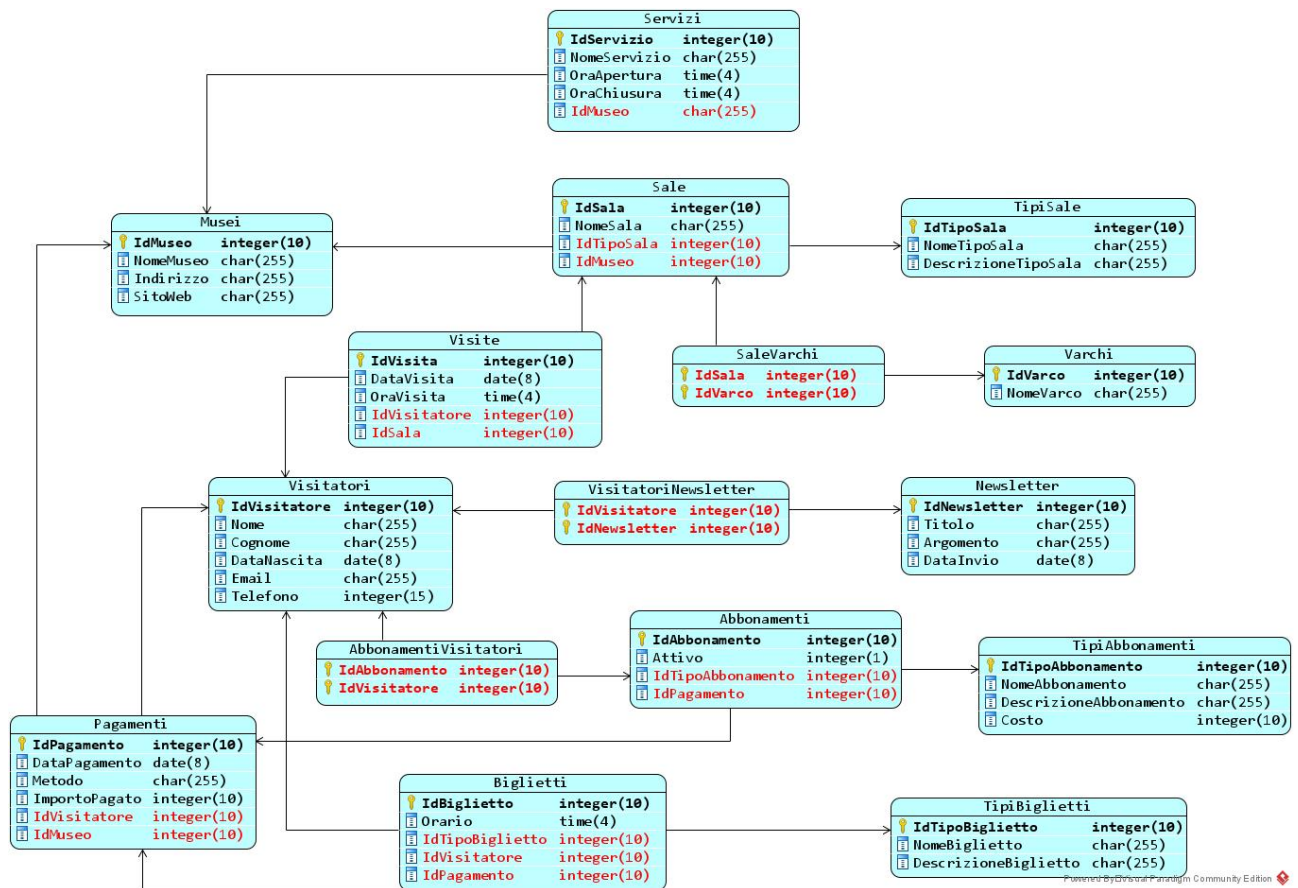
- In caso di visitatore con biglietto, il pagamento avviene alla fine della visita al museo. Di conseguenza la riga della tabella *Pagamenti* viene riempita in un secondo mo-

mento rispetto alla riga della tabella *Biglietti*. Ciò implica che la chiave esterna *Biglietti.IdPagamento* è NULL fino a quando non verrà pagato il relativo biglietto.

- Se un visitatore acquista n abbonamenti allora *Pagamenti.ImportoPagato* deve essere \geq di $n * \text{TipiAbbonamenti.Costo}$ degli abbonamenti a cui si riferisce.
- Un visitatore può accedere alle sale esclusive sse ha un abbonamento speciale attivo.

3 Schema logico relazionale

3.1 Formato grafico



3.2 Formato testuale

- **Musei**(IdMuseo, NomeMuseo, Indirizzo, SitoWeb)
- **Servizi**(IdServizio, NomeServizio, OraApertura, OraChiusura, IdMuseo*)
- **Sale**(IdSala, NomeSala, IdTipoSala*, IdMuseo*)
- **TipiSale**(IdTipoSala, NomeTipoSala, DescrizioneTipoSala)
- **SaleVarchi**(IdSala*, IdVarco*)
- **Varchi**(IdVarco, NomeVarco)
- **Visite**(IdVisita, DataVisita, OraVisita, IdVisitatore*, IdSala*)

- **Visitatori**(IdVisitatore, Nome, Cognome, DataNascita, Email, Telefono)
- **VisitatoriNewsletter**(IdVisitatore*, IdNewsletter*)
- **Newsletter**(IdNewsletter, Titolo, Argomento, DataInvio)
- **AbbonamentiVisitatori**(IdAbbonamento*, IdVisitatore*)
- **Abbonamenti**(IdAbbonamento, Attivo, IdTipoAbbonamento*, IdPagamento*)
- **TipiAbbonamenti**(IdTipoAbbonamento, NomeAbbonamento, DescrizioneAbbonamento, Costo)
- **Biglietti**(IdBiglietto, Orario, IdTipoBiglietto*, IdVisitatore*, IdPagamento*)
- **TipiBiglietti**(IdTipoBiglietto, NomeBiglietto, DescrizioneBiglietto)
- **Pagamenti**(IdPagamento, DataPagamento, Metodo, ImportoPagato, IdVisitatore*, IdMuseo*)

3.3 Dipendenze funzionali

Le tabelle *SaleVarchi*, *VisitatoriNewsletter* e *AbbonamentiVisitatori* sono in BCNF perché non hanno dipendenze non banali.

Le altre tabelle sono anch'esse in BCNF perché, come riportato di seguito, la parte sinistra di ogni dipendenza non banale è superchiave.

- **Musei**
 - $\text{IdMuseo} \rightarrow \text{NomeMuseo}, \text{Indirizzo}, \text{SitoWeb}$
 - $\text{SitoWeb} \rightarrow \text{IdMuseo}, \text{NomeMuseo}, \text{Indirizzo}$
- **Servizi**
 - $\text{IdServizio} \rightarrow \text{NomeServizio}, \text{OraApertura}, \text{OraChiusura}, \text{IdMuseo}$
- **Sale**
 - $\text{IdSala} \rightarrow \text{NomeSala}, \text{IdTipoSala}, \text{IdMuseo}$
- **TipiSale**
 - $\text{IdTipoSala} \rightarrow \text{NomeTipoSala}, \text{DescrizioneTipoSala}$
- **Varchi**
 - $\text{IdVarco} \rightarrow \text{NomeVarco}$
- **Visite**
 - $\text{IdVisita} \rightarrow \text{DataVisita}, \text{OraVisita}, \text{IdVisitatore}, \text{IdSala}$
 - $\text{DataVisita}, \text{OraVisita}, \text{IdVisitatore}, \text{IdSala} \rightarrow \text{IdVisita}$
- **Visitatori**
 - $\text{IdVisitatore} \rightarrow \text{Nome}, \text{Cognome}, \text{DataNascita}, \text{Email}, \text{Telefono}$

- Email \rightarrow IdVisitatore, Nome, Cognome, DataNascita, Telefono
- Telefono \rightarrow IdVisitatore, Nome, Cognome, DataNascita, Email

- **Newsletter**

- IdNewsletter \rightarrow Titolo, Argomento, DataInvio
- Titolo, Argomento, DataInvio \rightarrow IdNewsletter

- **Abbonamenti**

- IdAbbonamento \rightarrow Attivo, IdTipoAbbonamento, IdPagamento

- **TipiAbbonamenti**

- IdTipoAbbonamento \rightarrow NomeAbbonamento, DescrizioneAbbonamento, Costo

- **Biglietti**

- IdBiglietto \rightarrow Orario, IdTipoBiglietto, IdVisitatore, IdPagamento
- Orario, IdTipoBiglietto, IdVisitatore, IdPagamento \rightarrow IdBiglietto

- **TipiBiglietti**

- IdTipoBiglietto \rightarrow NomeBiglietto, DescrizioneBiglietto

- **Pagamenti**

- IdPagamento \rightarrow DataPagamento, Metodo, ImportoPagato, IdVisitatore, IdMuseo

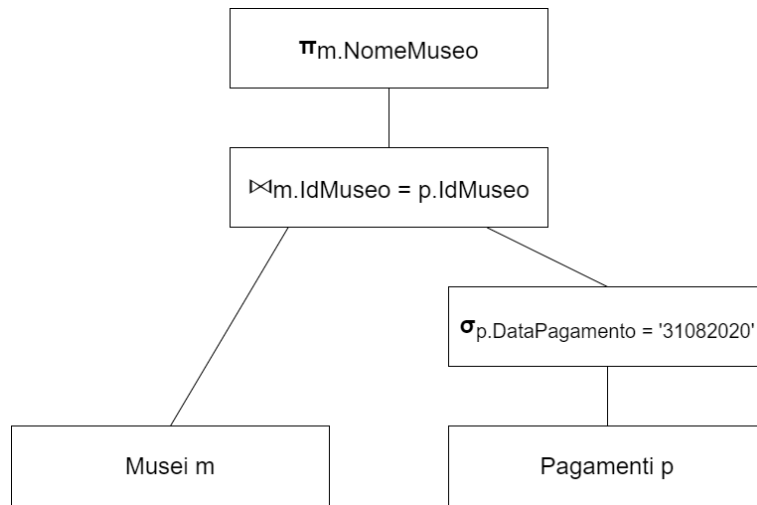
4 Interrogazioni in SQL

```
1  /* a. Uso di proiezione, join e restrizione:
2     Riportare il nome dei musei per ogni pagamento ricevuto in
3     data 31/08/2020.
4  */
5  SELECT m.NomeMuseo
6  FROM Musei m
7  JOIN Pagamenti p ON m.IdMuseo = p.IdMuseo
8  WHERE p.DataPagamento = '31082020';
9
10 /* b. Uso di group by con having, where e sort:
11     Per ogni metodo di pagamento, riportare in modo ordinato
12     la somma degli ImportoPagato > 20 se la somma è minore di 1000.
13  */
14 SELECT p.Metodo, SUM(p.ImportoPagato) as Somma
15 FROM Pagamenti p
16 WHERE p.ImportoPagato > 20
17 GROUP BY p.Metodo
18 HAVING SUM(p.ImportoPagato) < 1000
19 ORDER BY Somma;
20
21 /* c. Uso di join, group by con having e where:
22     Per ogni metodo di pagamento, riportare la somma degli
23     ImportoPagato > 20 dei visitatori che si chiamano Lorenzo
24     se la somma è minore di 1000.
25  */
26 SELECT p.Metodo, SUM(p.ImportoPagato)
27 FROM Pagamenti p
28 JOIN Visitatori v ON p.IdVisitatore = v.IdVisitatore
29 WHERE p.ImportoPagato > 20 AND v.Nome = 'Lorenzo'
30 GROUP BY p.Metodo
31 HAVING SUM(p.ImportoPagato) < 1000;
32
33 /* d. Uso di select annidata con quantificazione esistenziale:
34     Riportare tutti i dati dei visitatori che hanno effettuato
35     almeno un pagamento prima del 1/01/2021.
36  */
37 SELECT *
38 FROM Visitatori v
39 WHERE EXISTS (
40     SELECT *
41     FROM Pagamenti p
42     WHERE v.IdVisitatore = p.IdVisitatore AND p.DataPagamento < '01012021');
43
44 /* e. Uso di select annidata con quantificazione universale:
45     Riportare Id, nome e cognome dei visitatori che hanno sempre pagato
46     in contanti. Nel risultato non si vogliono vedere i dati di chi
47     non ha mai pagato.
48  */
49 SELECT v.IdVisitatore, v.Nome, v.Cognome
50 FROM Visitatori v
51 JOIN Pagamenti p1 ON v.IdVisitatore = p1.IdVisitatore
52 WHERE NOT EXISTS (
53     SELECT *
54     FROM Pagamenti p2
55     WHERE v.IdVisitatore = p2.IdVisitatore AND p2.Metodo <> 'Contanti');
56
57 /* f. Uso di subquery di confronto quantificato usando una subquery:
58     Riportare tutti i dati dei pagamenti effettuati con bancomat il
59     cui importo è maggiore di ogni pagamento effettuato in contanti.
60     NB: (n > ALL Set)  $\equiv (\forall x \in \text{Set} . x < n)$ 
61  */
62 SELECT *
63 FROM Pagamenti p1
64 WHERE p1.Metodo = 'Bancomat' AND p1.ImportoPagato > ALL (
65     SELECT p2.ImportoPagato
66     FROM Pagamenti p2
67     WHERE p2.Metodo = 'Contanti');
```

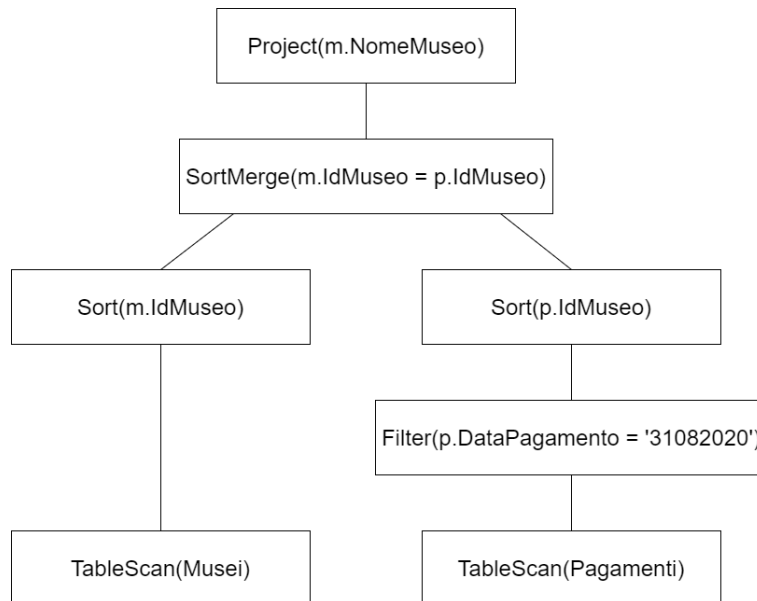
5 Piani di accesso

5.1 Query (a)

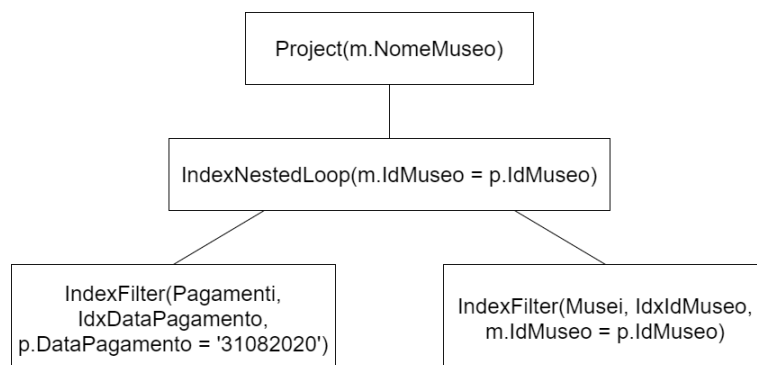
5.1.1 Piano di accesso logico



5.1.2 Piano di accesso fisico senza indici

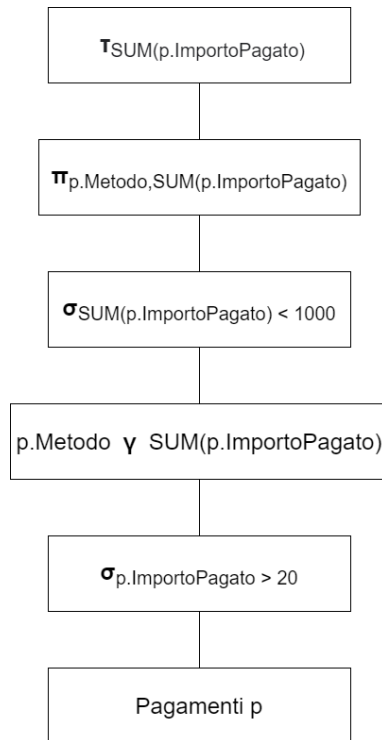


5.1.3 Piano di accesso fisico con indici

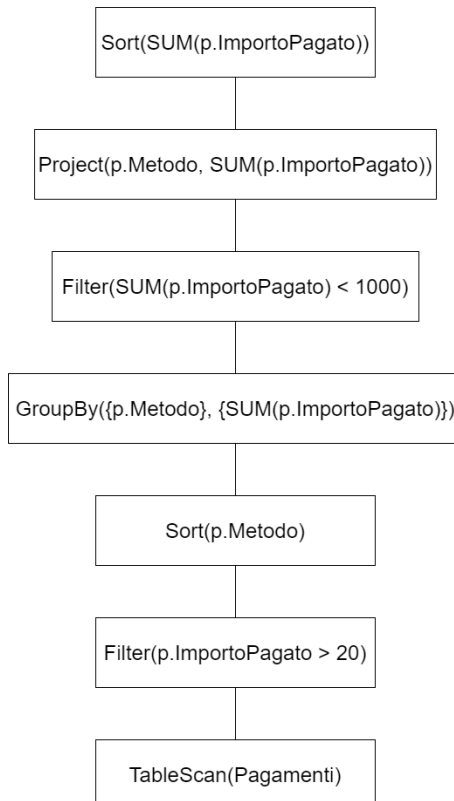


5.2 Query (b)

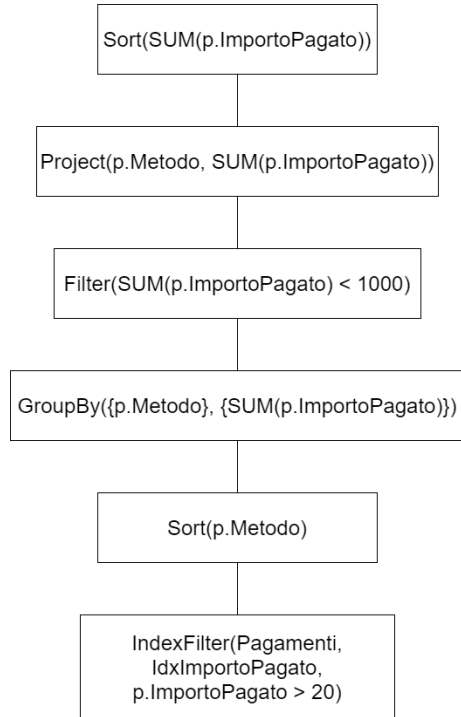
5.2.1 Piano di accesso logico



5.2.2 Piano di accesso fisico senza indici

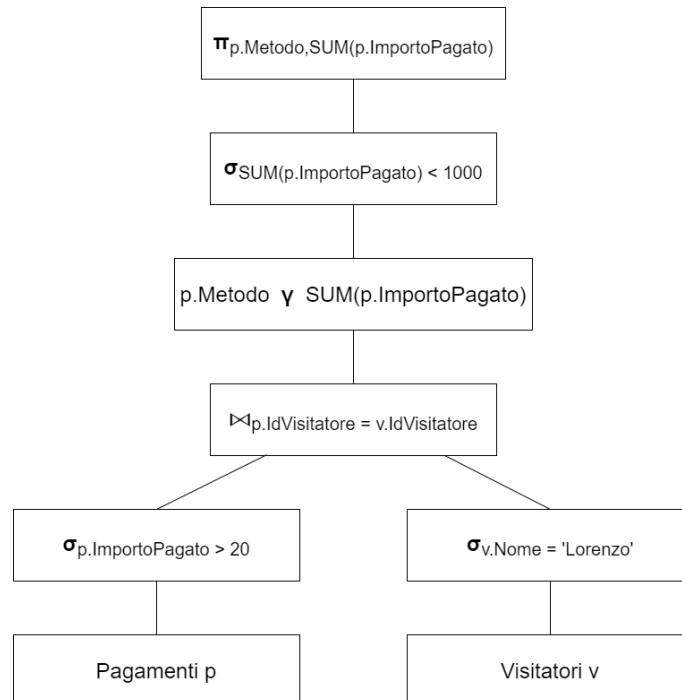


5.2.3 Piano di accesso fisico con indici

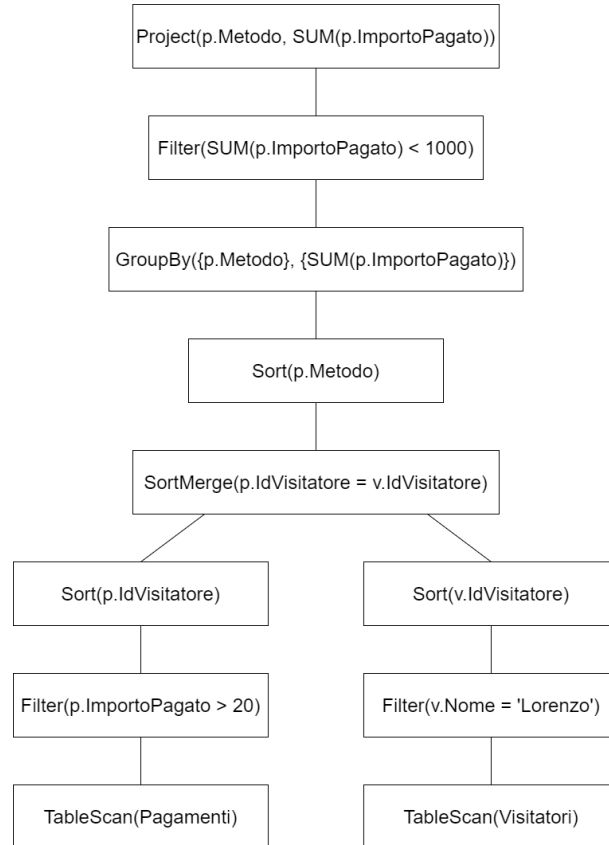


5.3 Query (c)

5.3.1 Piano di accesso logico



5.3.2 Piano di accesso fisico senza indici



5.3.3 Piano di accesso fisico con indici

