

UNIVERSITÀ DI PISA

PROGETTO BASI DI DATI

“Il museo segreto”

Basi di dati

Anno Accademico 21/22

Tommaso Somigli

Mat. 598357

Sommario

1.	Descrizione del dominio	3
2.	Schema concettuale	3
a.	Vincoli intrarelazionali	3
b.	Vincoli interrelazionali	4
3.	Schema logico relazionale.....	4
a.	Formato testuale	4
b.	Dipendenze funzionali	5
4.	Interrogazioni SQL	6
a.	Uso di proiezione, join e restrizione	6
b.	Uso di group by con having, where e sort	6
c.	Uso di join, group by con having e where.....	6
d.	Uso di select annidata con quantificazione esistenziale	6
e.	Uso di select annidata con quantificazione universale	6
f.	Uso di subquery di confronto quantificato usando una subquery.....	7
5.	Piani di Accesso.....	7
1.a.	Piano di accesso logico	7
1.b.	Piano di accesso logico.....	8
1.c.	Piano di accesso logico	9
2.a.	Piano di accesso fisico senza indici	10
2.b.	Piano di accesso fisico senza indici.....	11
2.c.	Piano di accesso fisico senza indici	12
3.a.	Piano di accesso fisico con indici	13
3.b.	Piano di accesso fisico con indici	13
3.c.	Piano di accesso fisico con indici	14

1. Descrizione del dominio

In **grassetto>** sono riportate le classi.

“Il Museo Segreto” è un’applicazione che permette di giocare alla caccia al tesoro all’interno di un museo mirata al divertimento dei visitatori.

- Registrazione dei visitatori
- Registrazione sui pagamenti effettuati
- Registrazione della tabella “giocata”

Il gioco parte all’acquisto (**Pagamenti**) dei biglietti (**Biglietti**) da parte dei visitatori (**Visitatori**).

Il gioco crea delle tabelle (**Tabelle**), a seconda del livello, suddivise in due tipologie:

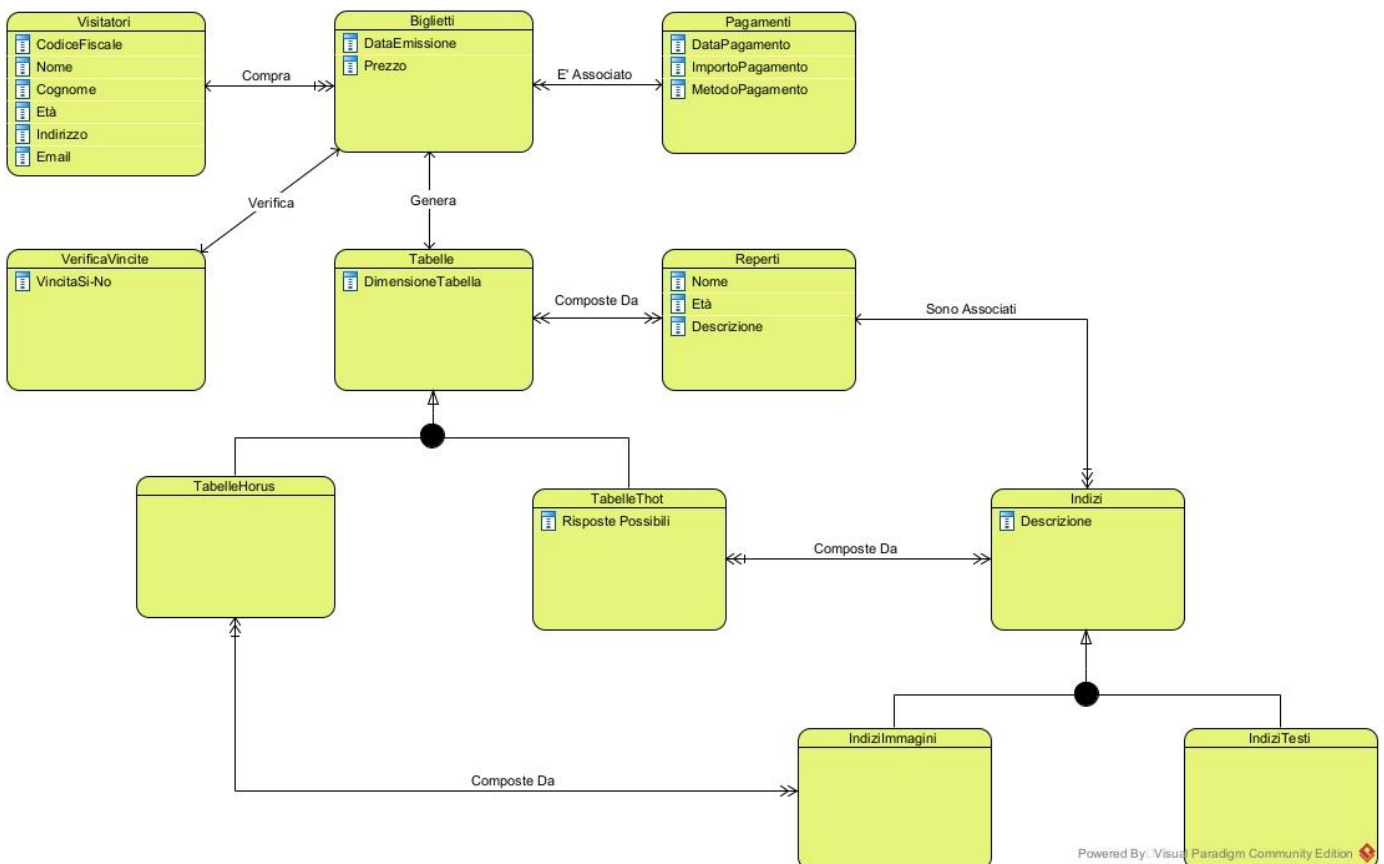
- Tabelle Horus (**TabelleHorus**)
- Tabelle Thot (**TabelleThot**)

La caccia al tesoro consiste nel trovare i reperti (**Reperti**) tramite degli indizi (**Indizi**) suddivisi in due tipologie:

- Indizi composti da immagini (**IndiziImmagini**)
- Indizi composti da testi (**IndiziTesti**)

La vincita è verificata alla consegna del biglietto e il sistema verificherà se è stata completata la tabella (**VerificaVincite**).

2. Schema concettuale



a. Vincoli intrarelazionali

Sono i vincoli che devono essere rispettati dai valori contenuti nella relazione considerata

- Se non specificato sotto, nessun attributo pu  valere NULL
- Se non specificato sotto, nessun attributo pu  valere NULL
- Non possono esistere due Persone con lo stesso codice fiscale

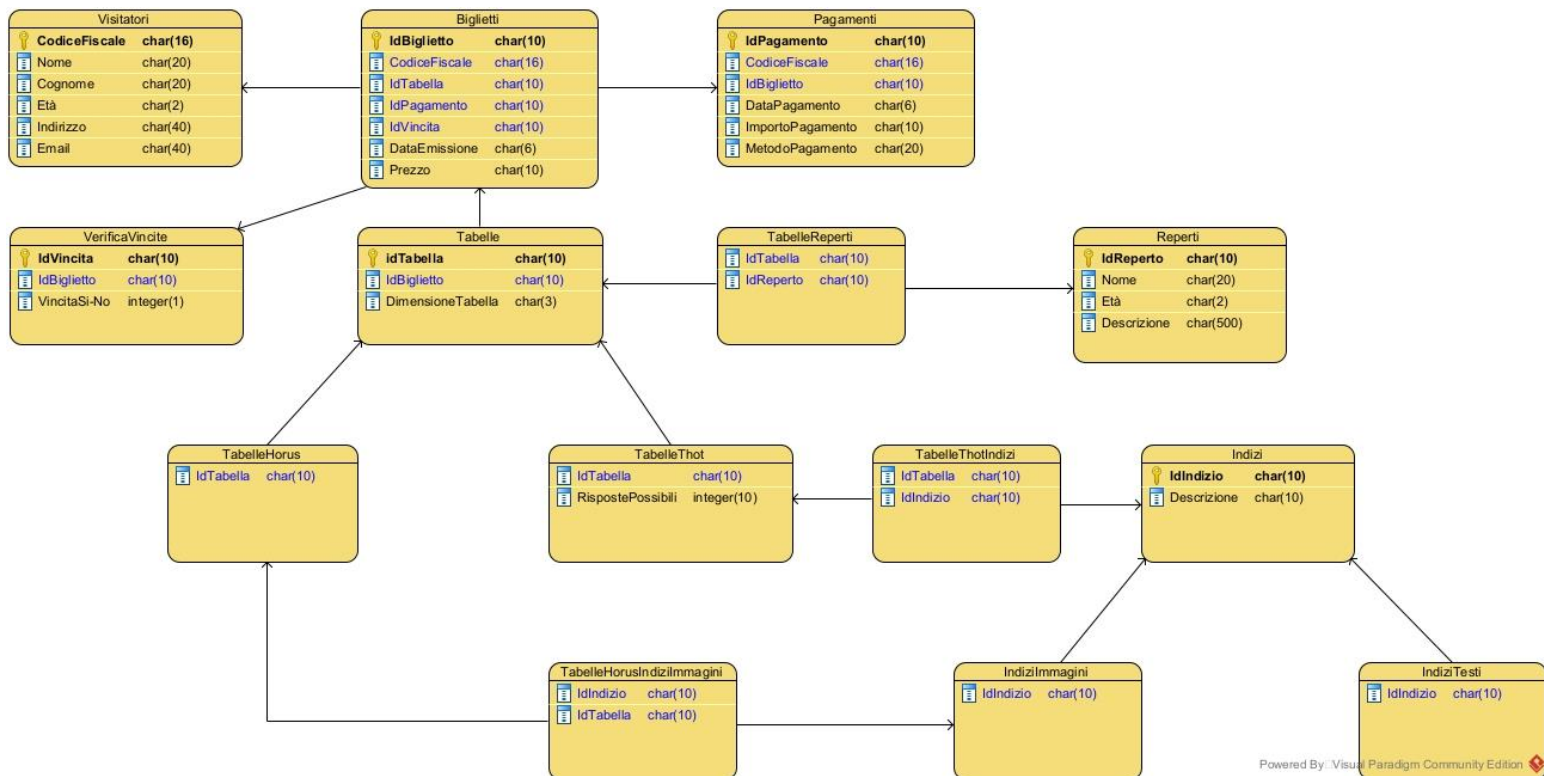
- L'attributo CodiceFiscale della classe Visitatori deve contenere esattamente 16 caratteri
- VincitàSi-No in VerificaVincite può assumere valori 0 o 1
- L'attributo ImportoPagato della classe Pagamenti deve essere > 0

b. Vincoli interrelazionali

Sono i vincoli che devono essere rispettati da valori contenuti in relazioni diverse

- Ogni biglietto deve essere associato ad una sola persona
- Ad ogni biglietto è associato un pagamento
- Ad ogni biglietto è associata un'eventuale vincita
- Ad ogni biglietto è associata una tabella
- Non possono esistere tabelle Horus che non sono tabelle
- Non possono esistere tabelle Thot che non sono tabelle
- Non possono esistere indizi formati da immagini immagini che non sono indizi
- Non possono esistere indizi formati da testi che non sono indizi

3. Schema logico relazionale



a. Formato testuale

Visitatori(CodiceFiscale, Nome, Cognome, Età, Indirizzo, Email)

Biglietti(IdBiglietto, CodiceFiscale*, IdPagamento*, IdVincita*, IdTabella*, DataEmissione, Prezzo)

Pagamenti(IdPagamento, CodiceFiscale*, IdBiglietto*, DataPagamento, ImportoPagamento, MetodoPagamento)

Tabelle(IdTabella, IdBiglietto*, DimensioneTabella)

TabelleHorus(IdTabella*)

TabelleThot(IdTabella*)

VerificaVincite(IdVincita, IdBiglietto*, VincitaSi-No)

Reperti(IdReperto, Nome, Età, Descrizione)

TabelleReperti(IdTabella*, IdReperto*)

Indizi(IdIndizio, Descrizione)

Indizilmmagini(IdIndizio*)

IndiziTesti(IdIndizio*)

TabelleThotIndizi(IdTabella*, IdIndizio*)

TabelleHorusIndizilmmagini(IdIndizio*, IdTabella*)

b. Dipendenze funzionali

Uno schema R, avente insieme di attributi T e insieme di dipendenze funzionali F, R, è in forma normale di Boyce-Codd (BCNF) se ogni dipendenza funzionale della chiusura di F o è banale, o ha come determinante una superchiave di T. Esiste un teorema che semplifica il calcolo, asserendo che se la condizione di cui sopra vale per una qualsiasi copertura di F allora vale per l'intera chiusura di F.

Tutte le relazioni rispettano la forma normale di Boyce-Codd (BCNF), di seguito si riportano le dipendenze funzionali di ciascuna:

- Le tabelle **TabelleReperti**, **TabelleHorus**, **TabelleThot**, **TabelleThotIndizi**, **Indizilmmagini**, **IndiziTesti**, **TabelleThotIndizilmmagini** sono in BCNF perché non hanno dipendenze non banali.
- **Visitatori**
 - CodiceFiscale → Nome, Cognome, Età, Indirizzo, Email
- **Biglietti**
 - IdBiglietto → IdPagamento, CodiceFiscale, IdTabella, IdVincita, CodiceFiscale, DataEmissione, Prezzo
 - IdTabella → IdPagamento, IdBiglietto, CodiceFiscale, IdTabella, IdVincita, DataEmissione, Prezzo
- **Pagamenti**
 - IdPagamento → CodiceFiscale, IdBiglietto, ImportoPagamento, DataPagamento, MetodoPagamento
- **VerificaVincite**
 - IdVincita → IdBiglietto, VincitaSi-No
 - IdBiglietto → IdVincita, VincitaSi-No
- **Tabelle**
 - IdTabella → IdBiglietto, DimensioniTabella
 - IdBiglietto → IdTabella, DimensioniTabella
- **Reperti**
 - IdReperto → Nome, Età, Descrizione
- **Indizi**
 - IdIndizio → Descrizione

4. Interrogazioni SQL

a. Uso di proiezione, join e restrizione

Riportare il nome di ogni persona per tutti i pagamenti ricevuti in data 17/06/2022

```
SELECT v.Nome
FROM Visitatori v
JOIN Pagamenti p ON v.CodiceFiscale = p.CodiceFiscale
WHERE p.DataPagamento = '17062022'
```

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

Riportare, per ogni visitatore che ha effettuato almeno 5 cacce al tesoro, il codice fiscale e il numero di cacce effettuate al livello Thot. Il tutto in ordine decrescente

```
SELECT t.CodiceFiscale, COUNT(t.IdTabella) as NumeroCacce
FROM Tabella t
WHERE t.DimensioniTabella = '5x5'
GROUP BY t.CodiceFiscale
HAVING COUNT(t.IdTabella) >= 5
ORDER BY COUNT(t.IdVisita) DESC
```

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

Riportare per ogni metodo di pagamento, la somma degli ImportoPagamento > 20 dei visitatori che si chiamano Carlo se la somma è minore di 1000

```
SELECT p.MetodoPagamento, SUM(p.ImportoPagamento) as Somma
FROM Pagamenti p
JOIN Visitatori v ON p.IdVisitatore = v.IdVisitatore
WHERE p.ImportoPagamento > 20 AND v.Nome = 'Carlo'
GROUP BY p.MetodoPagamento
HAVING SUM (p.ImportoPagamento) < 1000
```

d. Uso di select annidata con quantificazione esistenziale

Riportare tutti i dati dei visitatori che hanno effettuato almeno un pagamento prima del 17/06/2022

```
SELECT *
FROM Visitatori v
WHERE EXISTS (
    SELECT *
    FROM Pagamenti p
    WHERE v.CodiceFiscale = p.CodiceFiscale AND p.DataPagamento < '17062022' )
```

e. Uso di select annidata con quantificazione universale

Riportare l'id di tutti i biglietti che non sono vincenti

```
SELECT b.idBiglietto
FROM biglietti b
WHERE NOT EXISTS(
    SELECT*
```

```
FROM VerificaVincite vs
WHERE b.IdVincita = vs.IdVincita
```

f. Uso di subquery di confronto quantificato usando una subquery

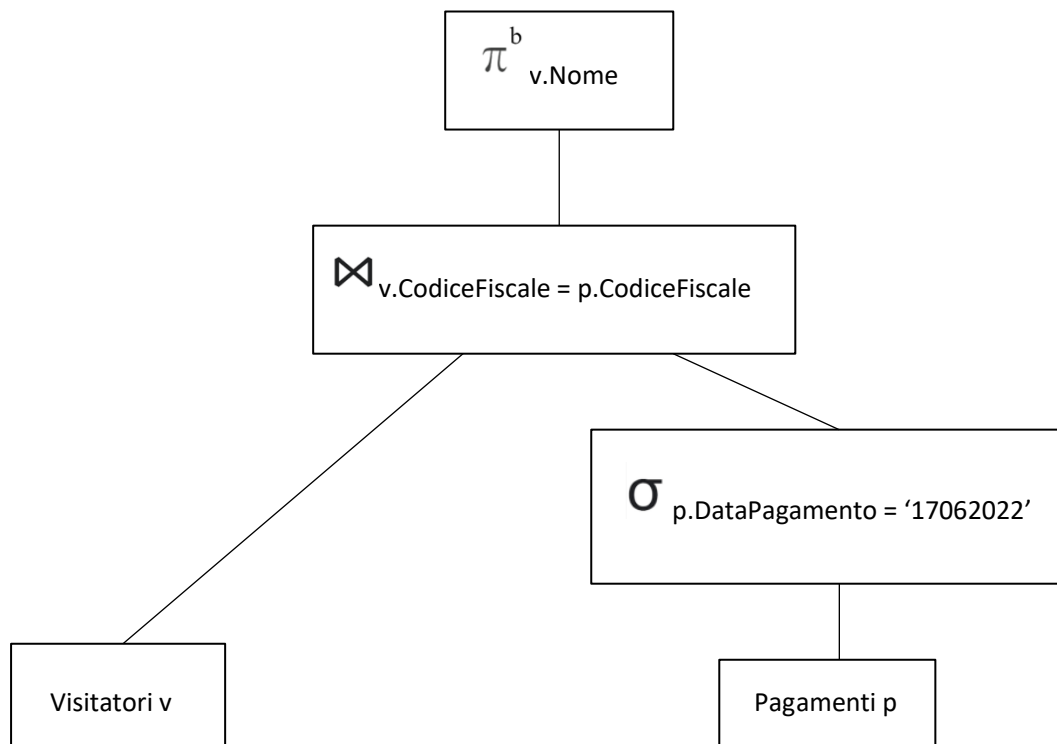
Riportare il codice fiscale e nome di tutti e soli gli utenti che hanno giocato almeno 2 cacce

```
SELECT v.CodiceFiscale, v.Nome
FROM Visitatori v
WHERE 2 <=
  ( SELECT COUNT(*)
    FROM Tabelle t
    WHERE ( t.CodiceFiscale = v.CodiceFiscale )
  )
```

5. Piani di Accesso

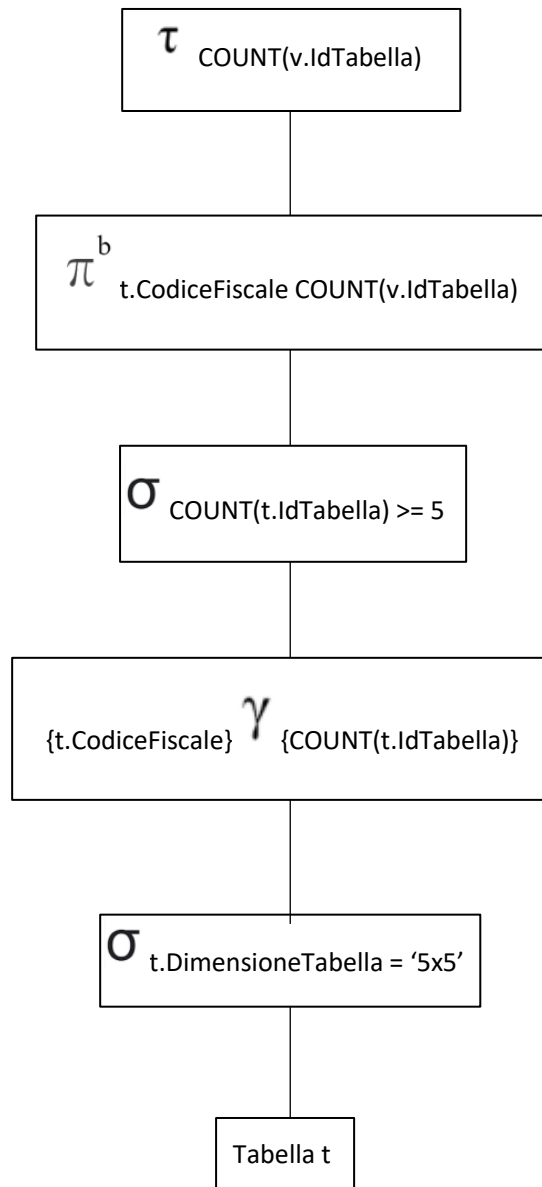
1.a. Piano di accesso logico

```
SELECT v.Nome
FROM Visitatori v
JOIN Pagamenti p ON v.CodiceFiscale = p.CodiceFiscale
WHERE p.DataPagamento = '17062022'
```



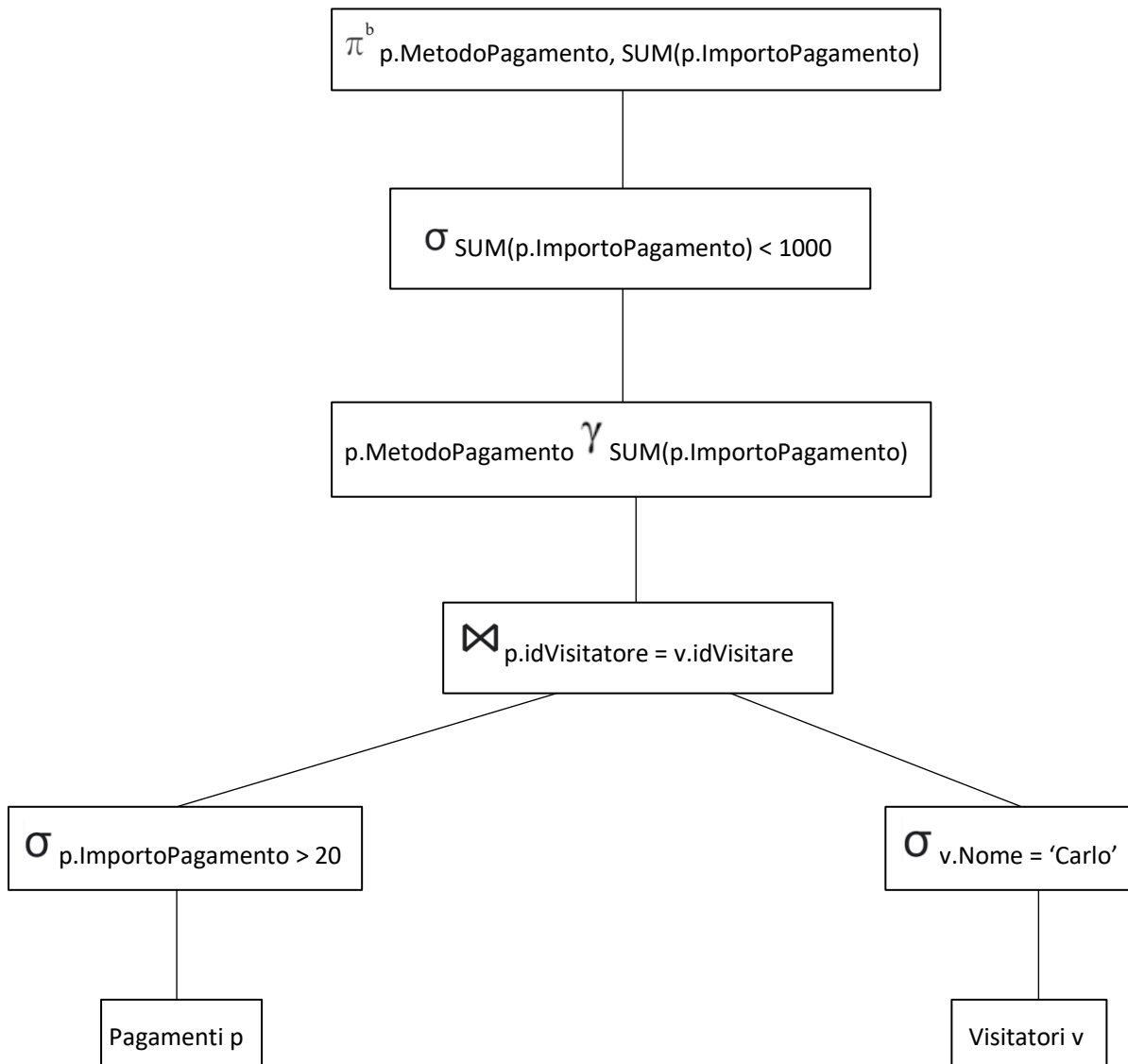
1.b. Piano di accesso logico

```
SELECT t.CodiceFiscale, COUNT(t.IdTabella) as NumeroCacce  
FROM Tabella t  
WHERE t.DimensioniTabella = '5x5'  
GROUP BY t.CodiceFiscale  
HAVING COUNT(t.IdTabella) >= 5  
ORDER BY COUNT(t.IdTabella) DESC
```

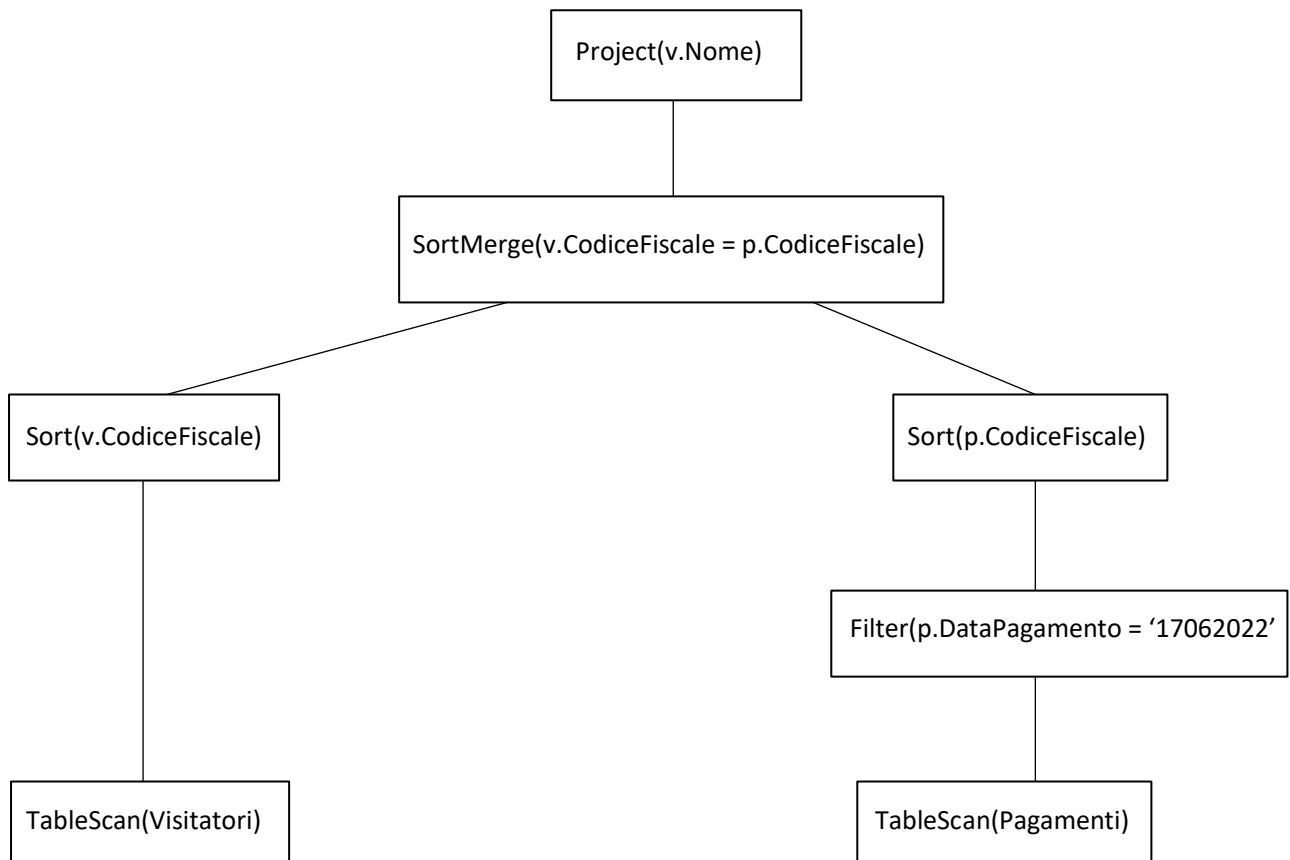


1.c. Piano di accesso logico

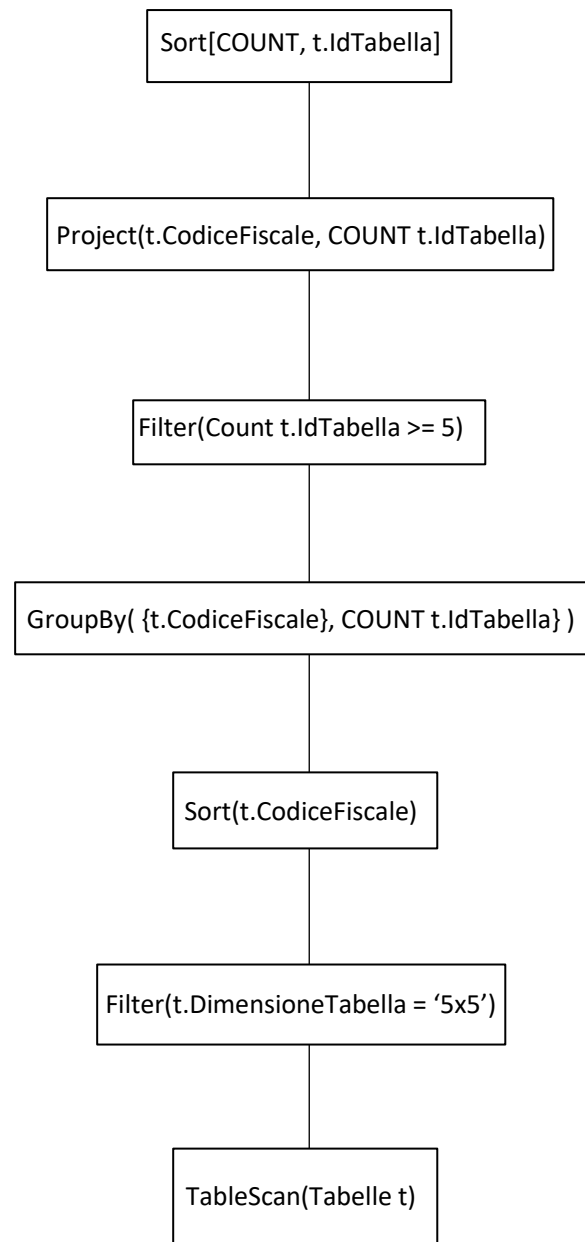
```
SELECT p.MetodoPagamento, SUM(p.ImportoPagamento) as Somma
FROM Pagamenti p
JOIN Visitatori v ON p.idVisitatore = v.idVisitatore
WHERE p.ImportoPagamento > 20 AND v.Nome = 'Carlo'
GROUP BY p.MetodoPagamento
HAVING SUM (p.ImportoPagamento) < 1000
```



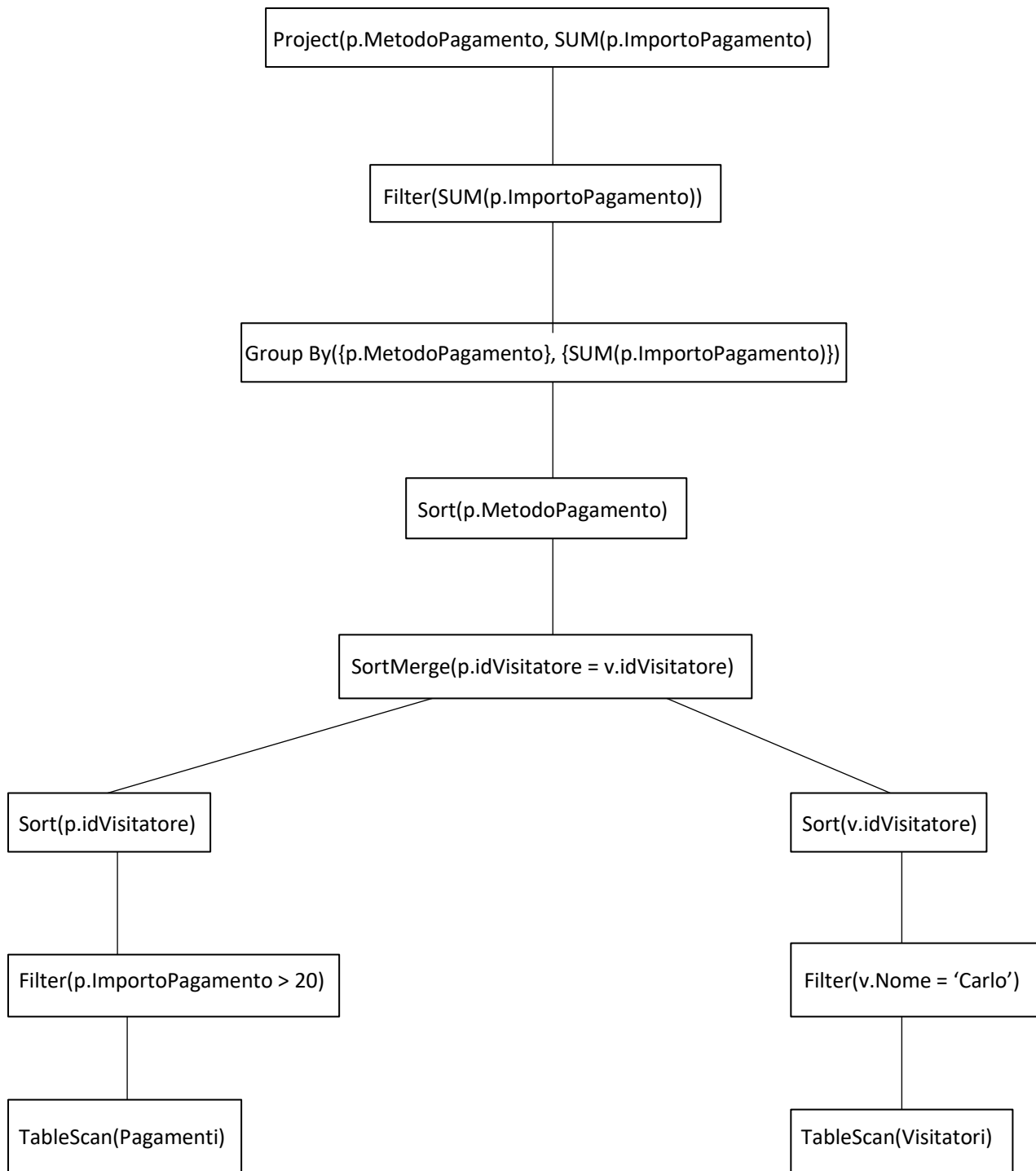
2.a. Piano di accesso fisico senza indici



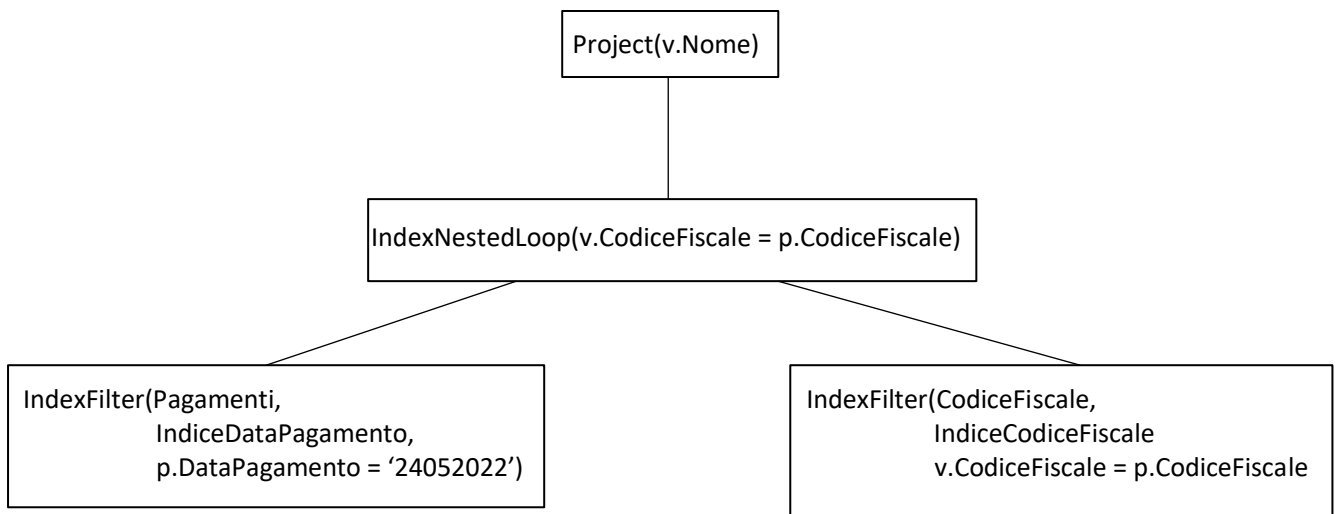
2.b. Piano di accesso fisico senza indici



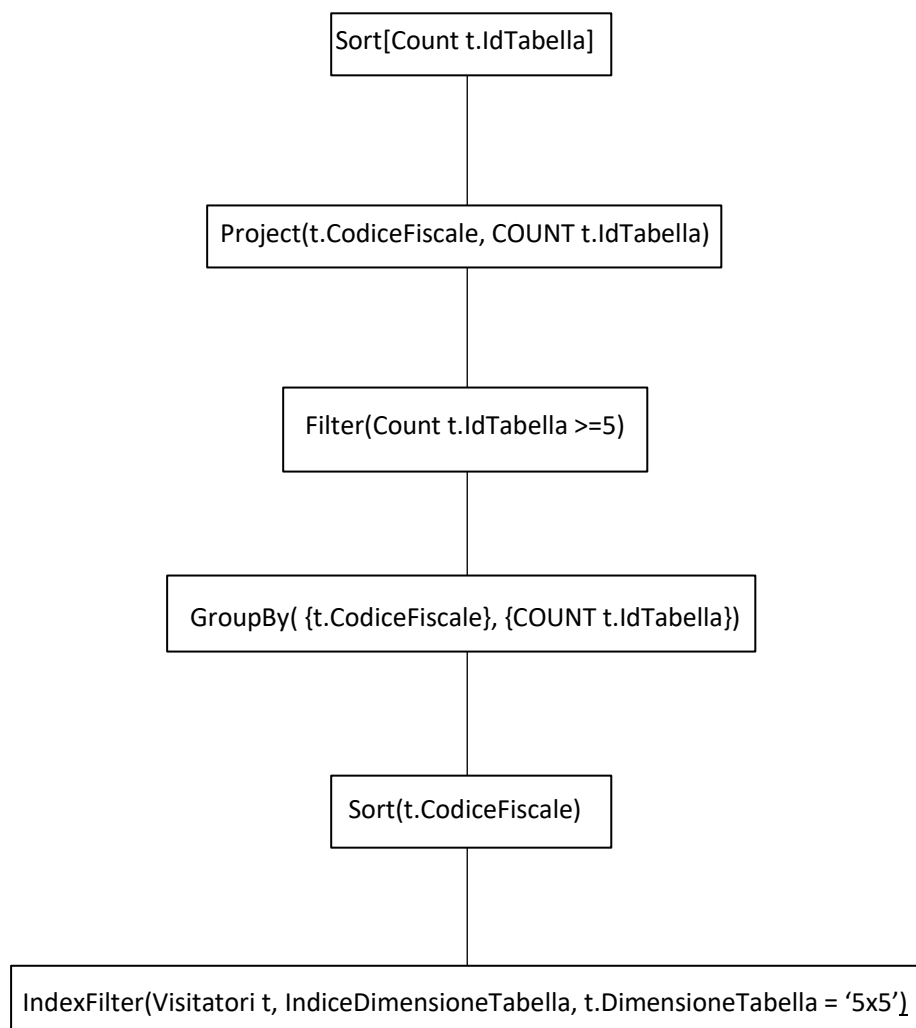
2.c. Piano di accesso fisico senza indici



3.a. Piano di accesso fisico con indici



3.b. Piano di accesso fisico con indici



3.c. Piano di accesso fisico con indici

