

UNIVERSITÀ
DI PISA

ANNO ACCADEMICO 2020-2021

BASI DI DATI

PROGETTO
"UN GIORNO AL MUSEO"

FABIO MARCHETTI - 581806
MATTEO CECCHERINI - 607086

05/06/2021



SOMMARIO

1. Descrizione del dominio	3
2. Schema concettuale	4
a. Vincoli intrarelazionali	5
b. Vincoli interrelazionali	5
3. Schema logico relazionale	5
a. Formato testuale	7
b. Dipendenze Funzionali	7
4. Interrogazioni SQL	10
a. Uso di proiezione, join e restrizione	10
b. Uso di group by con having, where e sort	10
c. Uso di join, group by con having e where	10
d. Uso di select annidata con quantificazione esistenziale	11
e. Uso di select annidata con quantificazione universale	11
f. Uso di subquery di confronto quantificato usando una subquery	11
5. Piani di accesso	12
1.a. Piano di accesso logico	12
1.b. Piano di accesso logico	13
1.c. Piano di accesso logico	14
2.a. Piano di accesso fisico senza indici	15
2.b. Piano di accesso fisico senza indici	16
2.c. Piano di accesso fisico senza indici	17
3.a. Piano di accesso fisico con indici	18
3.b. Piano di accesso fisico con indici	19
3.c. Piano di accesso fisico con indici	20

*sono indicate in grassetto le classi individuate

1. Descrizione del dominio

"Un giorno al museo" è un'applicazione che fornisce molteplici funzionalità, mirate a rendere più facile la gestione dei musei della regione Toscana:

- Servizio di newsletter attraverso l'indirizzo e-mail
- Registrazione dei pagamenti effettuati
- Registrazione delle informazioni sulle visite effettuate dai visitatori
- Rappresentazione logica della struttura fisica del museo (sale e varchi)
- Registrazione delle informazioni sui tipi di abbonamenti disponibili e sugli abbonamenti sottoscritti.
- Erogazione delle informazioni relative alle sale e alle opere all'interno del museo

Le informazioni relative alle opere sono salvate su un database esterno al sistema "DBOpere", e potranno essere accedute attraverso i dispositivi. DBOpere sarà comune a tutti i musei della regione Toscana per facilitare la condivisione delle informazioni in caso di mostre, in cui le opere vengono esposte in più musei.

Il museo è composto da varie sale (**Sale**), aventi una descrizione accessibile ai visitatori attraverso un codice QR. Le sale possono essere di diversi tipi:

- Sale museali
- Sale mostre
- Sale esclusive

Le varie sale sono connesse tra loro tramite dei varchi (**Varchi**). Si suppone che il gestore del museo possa voler conoscere, per un particolare varco, il numero di visitatori che lo attraversano.

Chiunque (**Persone**) sarà in grado di iscriversi alla newsletter (**IscrizioniNewsletter**), non sarà per forza necessario essere un visitatore. Sarà possibile farlo recandosi alla biglietteria, oppure grazie ad un'apposita pagina web sul sito del museo. Chiunque si iscriverà alla newsletter sarà registrato nella base di dati col suo codice fiscale ed il suo indirizzo email, al quale saranno spedite le **News** (anch'esse salvate sulla base di dati).

I dati relativi a un visitatore (**Visitatori**) vengono salvati al momento dell'inizializzazione della prima visita all'interno del museo, oppure al momento dell'acquisto di un abbonamento (**Abbonamento**). Ad ogni abbonamento saranno associati più visitatori, che saranno specificati dall'acquirente. I dati relativi al pagamento di un abbonamento saranno salvati sulla base di dati.

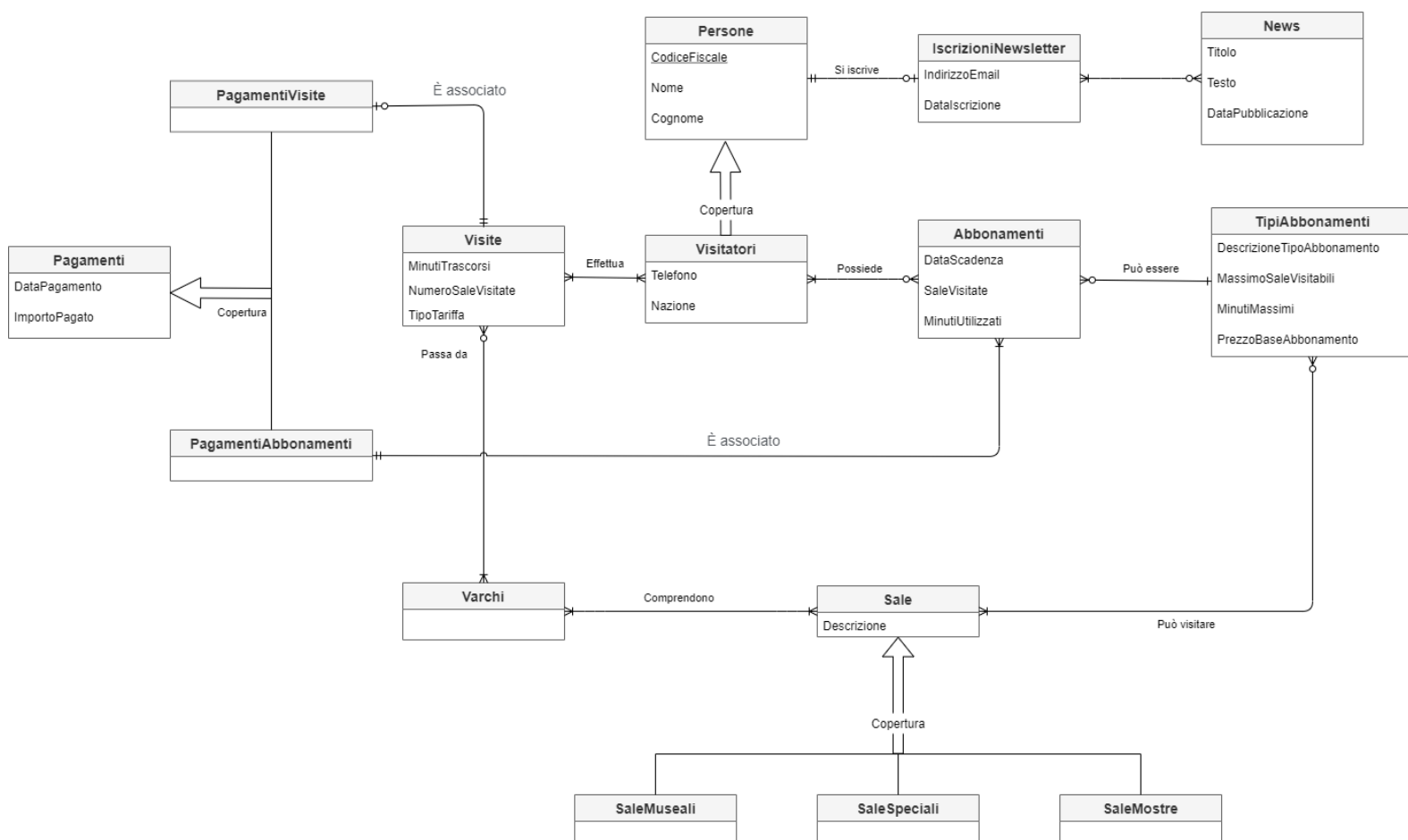
Il sistema mette a disposizione dei visitatori una scelta di vari tipi di abbonamento (**TipiAbbonamenti**), alcuni dei quali permetteranno di accedere a determinate sale speciali.

[TORNA AL SOMMARIO](#)

Tutti i pagamenti (**Pagamenti**) relativi al museo vengono registrati nel database e comprendono l'importo e la data in cui il pagamento è stato effettuato.

Quando un visitatore vuole effettuare una visita (**Visite**) seleziona il tipo di tariffa (tra le due disponibili) oppure si identifica, se abbonato (e l'abbonamento è in corso di validità). A quel punto, sulla base della tariffa scelta/abbonamento, il dispositivo viene inizializzato nei suoi registri interni. Durante la visita, lo stato interno del dispositivo viene aggiornato, sulla base delle azioni del visitatore. Alla fine della visita, vengono salvate sulla base di dati le varie informazioni raccolte dal dispositivo e viene registrato il pagamento della visita.

2. Schema concettuale



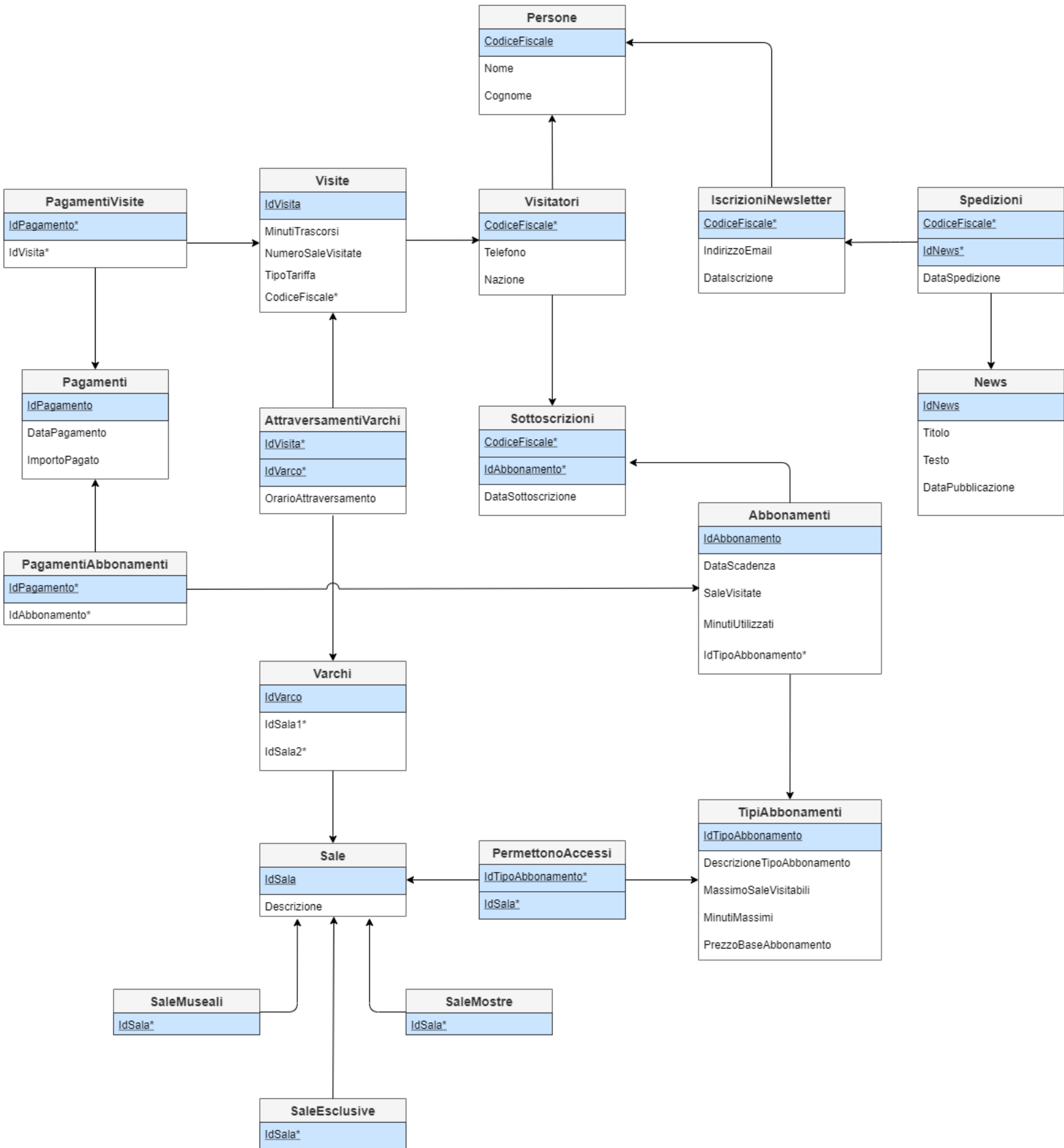
a. Vincoli intrarelazionali

- Se non specificato sotto, nessun attributo può valere NULL
- Non possono esistere due Persone con lo stesso codice fiscale.
- L'attributo CodiceFiscale della classe Persone deve contenere esattamente 16 caratteri
- Non possono esistere due Persone con la stessa email.
- TipoTariffa in Visite può assumere valori pari a:
 - "Bianca"
 - "Verde"
 - "Abbonamento"
- Non possono esistere due Visitatori con lo stesso numero di telefono.
- L'attributo telefono della classe Visitatori deve essere maggiore di 9 caratteri.
- L'attributo email della classe IscrizioniNewsletter deve contenere un carattere @ e almeno un punto.
- L'attributo SaleVisitate della classe Abbonamenti deve essere ≥ 0
- L'attributo MinutiUtilizzati della classe Abbonamenti deve essere ≥ 0
- L'attributo MassimoSaleVisitabili della classe TipiAbbonamenti deve essere > 0
- L'attributo MinutiMassimi della classe TipiAbbonamenti deve essere > 0
- L'attributo ImportoPagato della classe Pagamenti deve essere > 0
- L'attributo MinutiTrascorsi della classe Visite deve essere > 0
- L'attributo SaleVisitate della classe Visite deve essere > 0

b. Vincoli interrelazionali

- TipoTariffa non deve avere il valore "Abbonamento" se il visitatore non ha un abbonamento in corso di validità.
- Una visita non può essere associata a un varco che porta a una sala non compresa nel tipo di abbonamento attivo per quel visitatore.
- Ogni IscrizioneNewsletter deve appartenere a una sola persona.
- Ad ogni sala è associata a uno o più varchi.
- Le stesse due sale possono essere congiunte da più varchi.
- Ogni PagamentoVisita deve riferirsi a una visita.
- Ogni PagamentoAbbonamento deve riferirsi a un abbonamento.
- Non possono esistere Visitatori che non sono Persone.
- Non possono esistere PagamentiAbbonamenti che non sono Pagamenti.
- Non possono esistere PagamentiVisite che non sono Pagamenti.
- Non possono esistere SaleMuseali che non sono Sale.
- Non possono esistere SaleSpeciali che non sono Sale.
- Non possono esistere SaleMostre che non sono Sale.

3. Schema logico relazionale



a. Formato testuale

Persone(CodiceFiscale, Nome, Cognome)
IscrizioniNewsletter(CodiceFiscale*, IndirizzoEmail, DataIscrizione)
Spedizioni(CodiceFiscale*, IdNews*, DataSpedizione)
News(IdNews, Titolo, Testo, DataPubblicazione)
Visitatori(CodiceFiscale*, Telefono, Nazione)
Visite(IdVisita, MinutiTrascorsi, NumeroSaleVisitate, TipoTariffa)
Sottoscrizioni(CodiceFiscale*, IdAbbonamento, DataSottoscrizione)
Abbonamenti(IdAbbonamento, DataScadenza, IdTipoAbbonamento*,
SaleVisitate, MinutiUtilizzati)
AttraversamentiVarchi(IdVisita*, IdVarco*, OrarioAttraversamento)
PagamentiVisite(IdPagamento*, IdVisita*)
Pagamenti(IdPagamento, DataPagamento, ImportoPagato)
PagamentiAbbonamenti(IdPagamento*, IdAbbonamento*)
Varchi(IdVarco, Sala1*, Sala2*)
Sale(IdSala, Descrizione)
PermettonoAccessi(IdTipoAbbonamento*, IdSala*)
TipiAbbonamenti(IdTipoAbbonamento, DescrizioneTipoAbbonamento,
MassimoSaleVisitabili, MinutiMassimi, PrezzoBaseAbbonamento)
SaleMuseali(IdSala*)
SaleEsclusive(IdSala*)
SaleMostre(IdSala*)

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:

- **Persone** → è in BCNF perché *CodiceFiscale* è chiave.
 - *CodiceFiscale* → *Nome*, *Cognome*
- **IscrizioniNewsletter** → è in BCNF perché *CodiceFiscale* è chiave, e *IndirizzoEmail* è relativo a una persona sola, quindi è una chiave
 - *CodiceFiscale* → *IndirizzoEmail*, *DataIscrizione*
 - *IndirizzoEmail* → *CodiceFiscale*, *DataIscrizione*
- **Spedizioni** → è in BCNF perché *CodiceFiscale* e *IdNews* sono chiavi
 - *CodiceFiscale*, *IdNews* → *DataSpedizione*

- **News** → è in BCNF perché *IdNews* è chiave.
 - *IdNews* → *Titolo, Testo, DataPubblicazione*
- **Visitatori** → è in BCNF perché *CodiceFiscale* è chiave, e *Telefono* è relativo a un solo visitatore quindi è chiave
 - *CodiceFiscale* → *Telefono, Nazione*
 - *Telefono* → *CodiceFiscale, Nazione*
- **Visite** → è in BCNF perché *IdVisita* è chiave, e *CodiceFiscale* identifica una persona in maniera univoca, quindi è una chiave
 - *IdVisita* → *MinutiTrascorsi, NumeroSaleVisitate, TipoTariffa, CodiceFiscale*
 - *CodiceFiscale* → *IdVisita, MinutiTrascorsi, NumeroSaleVisitate, TipoTariffa*
- **Sottoscrizioni** → è in BCNF perché *CodiceFiscale* e *IdNews* sono chiavi
 - *CodiceFiscale, IdAbbonamento* → *DataSottoscrizione*
- **Abbonamenti** → è in BCNF perché *IdAbbonamento* identifica univocamente un abbonamento, e *IdTipoAbbonamento* è una chiave.
 - *IdAbbonamento* → *DataScadenza, SaleVisitate, MinutiUtilizzati, IdTipoAbbonamento*
 - *IdTipoAbbonamento* → *IdAbbonamento, DataScadenza, SaleVisitate, MinutiUtilizzati*
- **AttraversamentiVarchi** → è in BCNF perché *IdVisita* e *IdVarco* sono chiavi
 - *IdVisita, IdVarco* → *OrarioAttraversamento*
- **PagamentoVisite** → è in BCNF perché non ha dipendenze non banali.
- **PagamentoAbbonamenti** → è in BCNF perché non ha dipendenze non banali.
- **Varchi** → è in BCNF perché non ha dipendenze non banali.
- **PermettonoAccessi** → è in BCNF perché non ha dipendenze non banali.
- **SaleMuseali** → è in BCNF perché non ha dipendenze non banali.
- **SaleMostre** → è in BCNF perché non ha dipendenze non banali.
- **SaleEsclusive** → è in BCNF perché non ha dipendenze non banali.
- **Pagamenti** → è in BCNF perché *IdPagamento* è chiave

- *IdPagamento* → *DataPagamento, ImportoPagato*
- *Sale* → è in BCNF perché *IdSala* è chiave
 - *IdSala* → *Descrizione*
- *TipiAbbonamento* → è in BCNF perché *IdTipoAbbonamento* è chiave.
 - *IdTipoAbbonamento* → *DescrizioneTipoAbbonamento, MassimoSaleVisitabili, MinutiMassimi, PrezzoBaseAbbonamento*

4. Interrogazioni SQL

a. Uso di proiezione, join e restrizione

Query che restituisce l'id e il tipo della tariffa relativi alle visite al museo effettuate da tutti i visitatori francesi

```
SELECT vi.IdVisita, p.TipoTariffa
FROM Visitatori v
JOIN Visite vi ON (vi.IdVisitatore = v.IdVisitatore)
WHERE v.Nazione = "Francia";
```

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

Query che restituisce, per ogni visitatore che ha effettuato almeno 5 visite, l'IdVisitatore e il numero di visite da più di 30 minuti che ha effettuato. Il tutto, in ordine discendente di visite effettuate.

```
SELECT vi.IdVisitatore, COUNT(vi.IdVisita) as NumeroVisite
FROM Visite vi
WHERE vi.MinutiTrascorsi > 30
GROUP BY vi.IdVisitatore
HAVING COUNT(vi.IdVisita) >= 5
ORDER BY COUNT(vi.IdVisita) DESC;
```

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

Query che restituisce, per ogni tipo di abbonamento che permetta di visitare almeno 50 sale, il numero di abbonamenti sottoscritti che hanno tale tipo, se ne sono stati sottoscritti almeno 5.

```
SELECT ta.IdTipoAbbonamento,
COUNT(a.IdAbbonamento) as NumeroAbbonamenti
FROM Abbonamenti a
JOIN TipiAbbonamenti ta
ON (a.IdTipoAbbonamento = ta.IdTipoAbbonamento)
WHERE ta.MassimoSaleVisitabili >= 50
GROUP BY ta.IdTipoAbbonamento
HAVING COUNT(a.IdAbbonamento) >= 5;
```

d. Uso di select annidata con quantificazione esistenziale

Query che restituisce l'id delle news che siano state spedite ad almeno una persona.

```
SELECT n.IdNews, n.Titolo
FROM News n
WHERE EXISTS (
  SELECT *
  FROM Spedizioni s
  WHERE s.IdNews = n.IdNews
);
```

e. Uso di select annidata con quantificazione universale

Query che restituisce l'id di tutti i tipi di abbonamento che non permettono l'accesso alle sale speciali.

```
SELECT pa.IdTipoAbbonamento
FROM PermettonoAccessi pa
WHERE NOT EXISTS (
  SELECT *
  FROM SaleEsclusive se
  WHERE pa.IdSala = se.IdSala
);
```

f. Uso di subquery di confronto quantificato usando una subquery

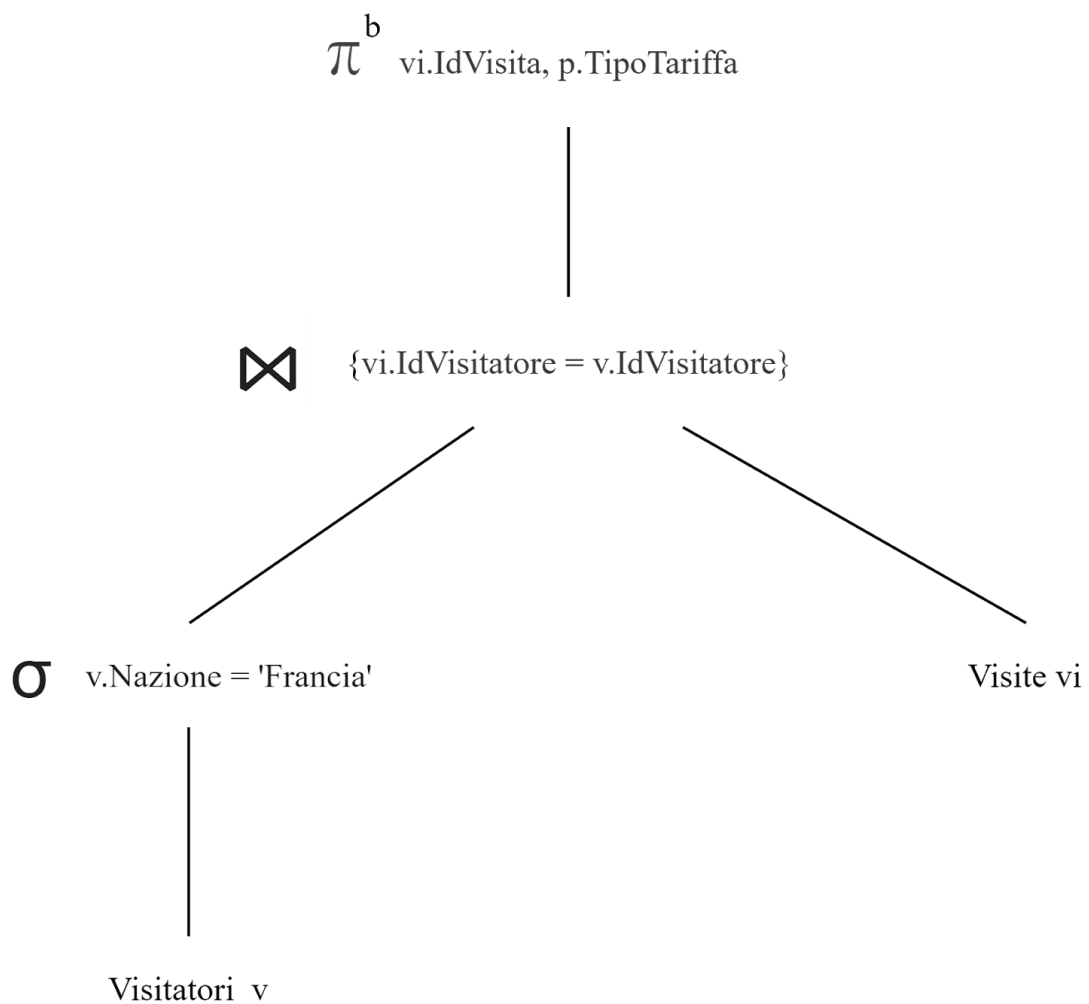
Query che restituisce la lista degli iscritti a cui non e' stata ancora spedita l'ultima news (sapendo che l'id più grande corrisponde alla news più recente)

```
SELECT isc.CodiceFiscale
FROM IscrizioniNewsletter isc
WHERE isc.IdNews < (
  SELECT MAX(n.IdNews)
  FROM News n
);
```

5. Piani di accesso

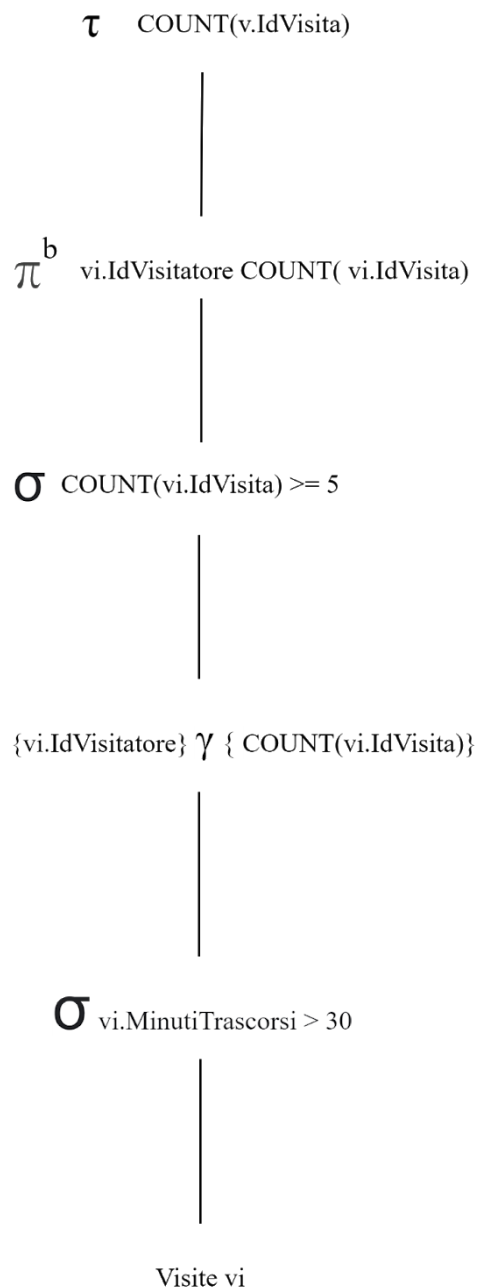
1.a. Piano di accesso logico

```
SELECT vi.IdVisita, p.TipoTariffa
FROM Visitatori v
JOIN Visite vi ON (vi.IdVisitatore = v.IdVisitatore)
WHERE v.Nazione = "Francia";
```



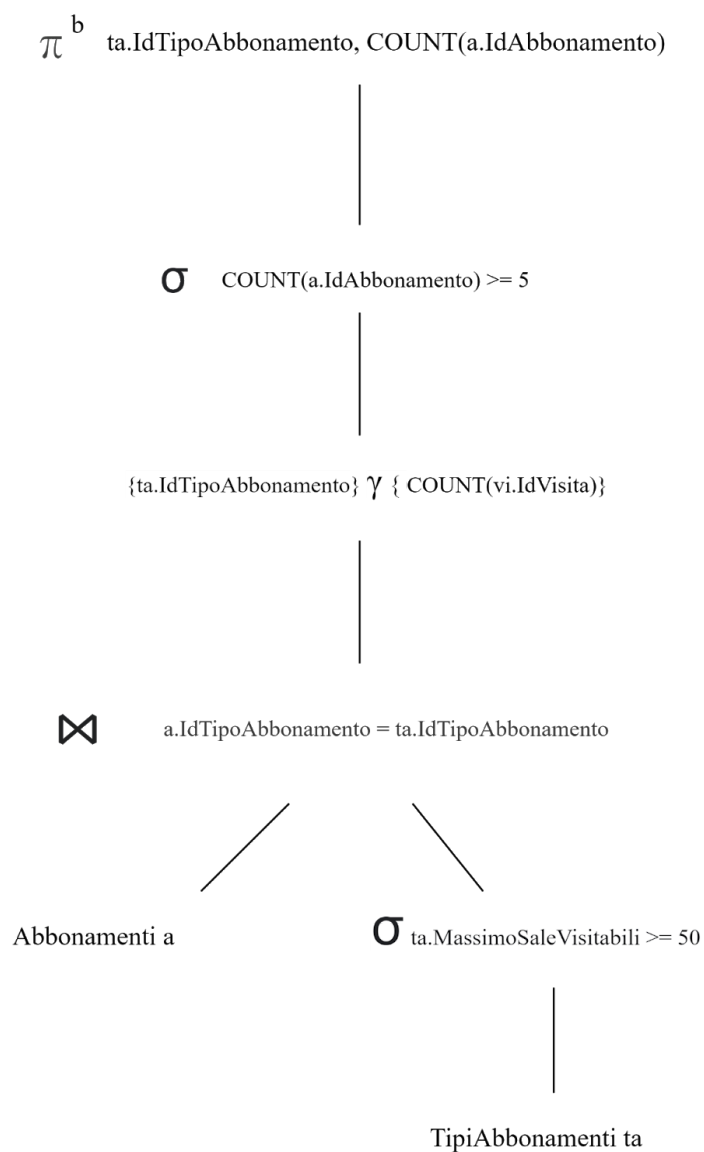
1.b. Piano di accesso logico

```
SELECT vi.IdVisitatore, COUNT(v.IdVisita) as NumeroVisite
FROM Visite vi
WHERE vi.MinutiTrascorsi > 30
GROUP BY vi.IdVisitatore
HAVING COUNT(vi.IdVisita) >= 5
ORDER BY COUNT(vi.IdVisita) DESC;
```

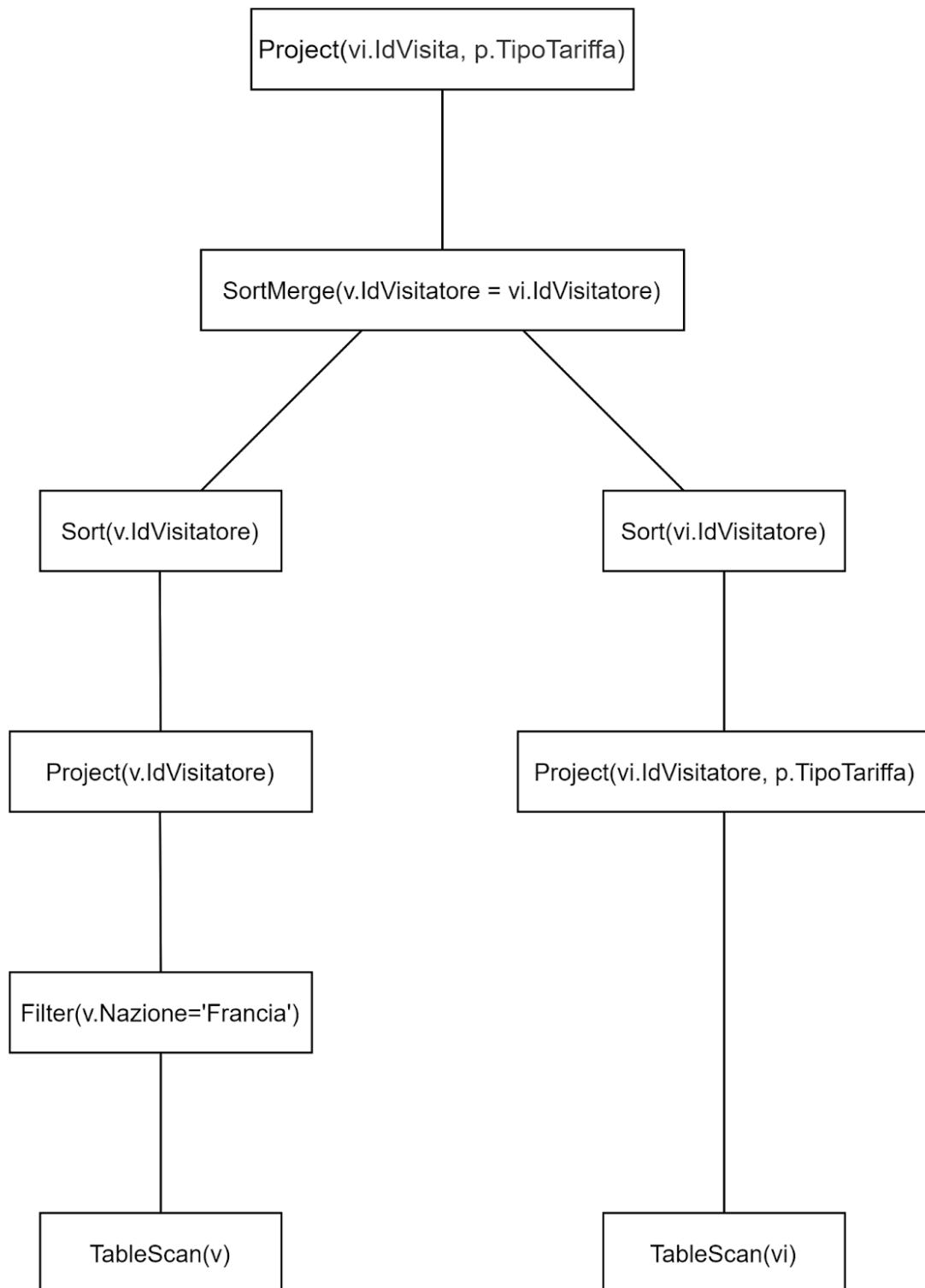


1.c. Piano di accesso logico

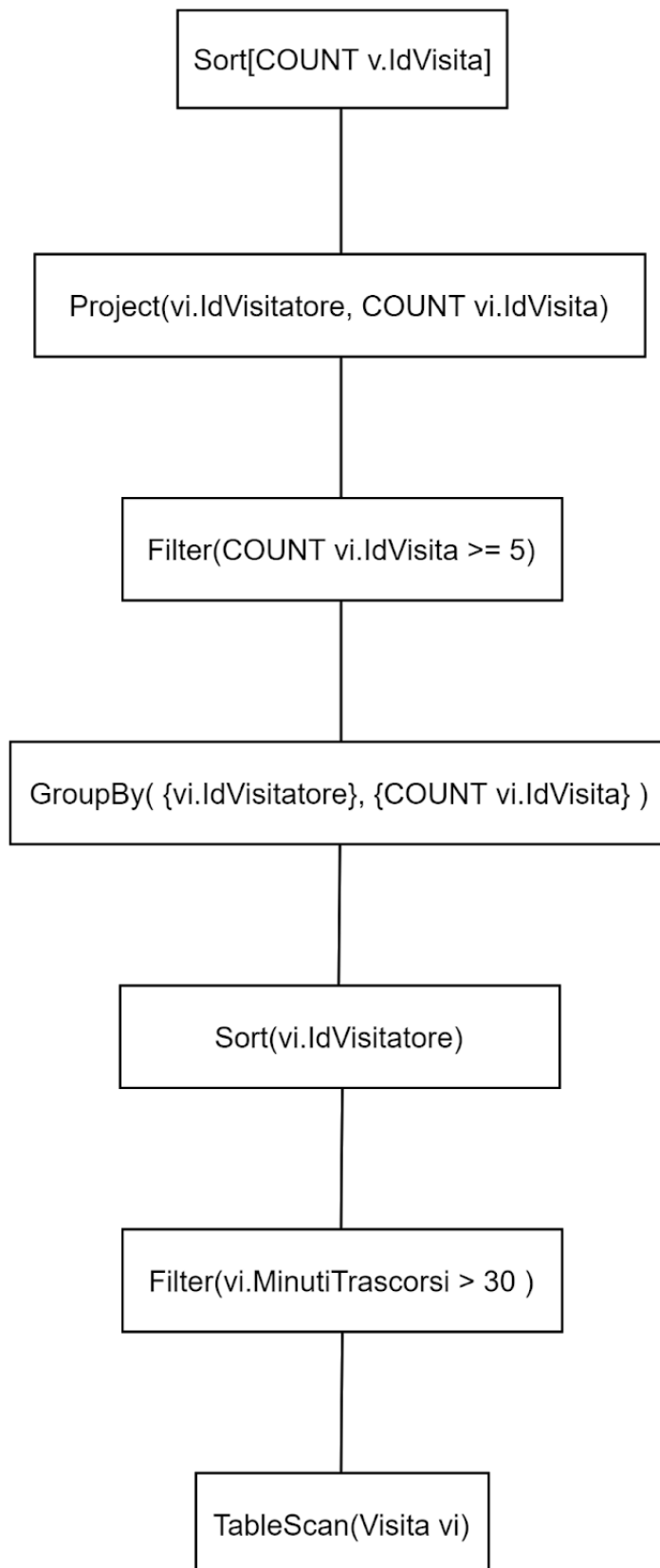
```
SELECT ta.IdTipoAbbonamento, COUNT(a.IdAbbonamento) as  
NumeroAbbonamenti  
FROM Abbonamenti a  
JOIN TipiAbbonamenti ta  
ON (a.IdTipoAbbonamento = ta.IdTipoAbbonamento)  
WHERE ta.MassimoSaleVisitabili >= 50  
GROUP BY ta.IdTipoAbbonamento  
HAVING COUNT(a.IdAbbonamento) >= 5;
```



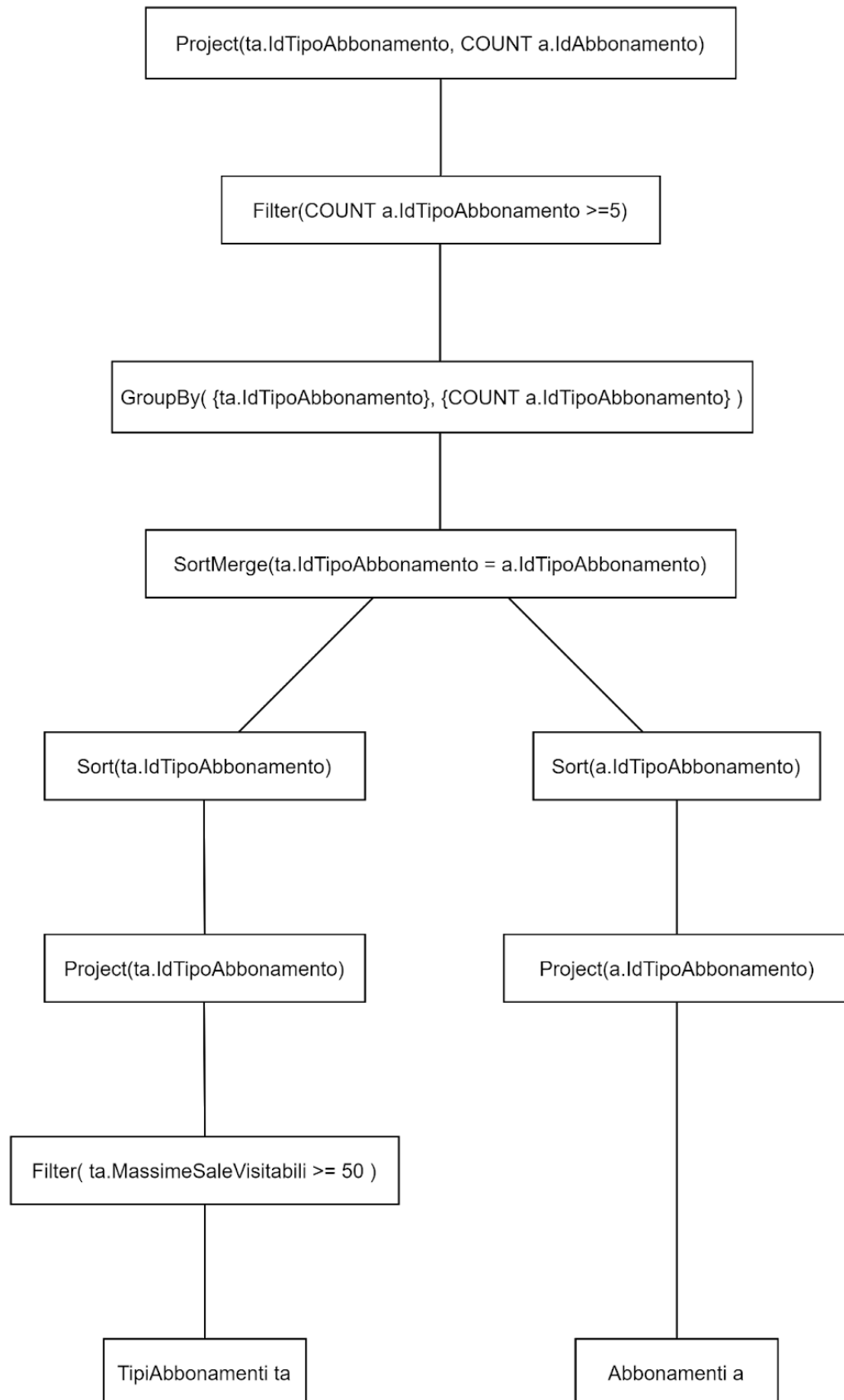
2.a. Piano di accesso fisico senza indici



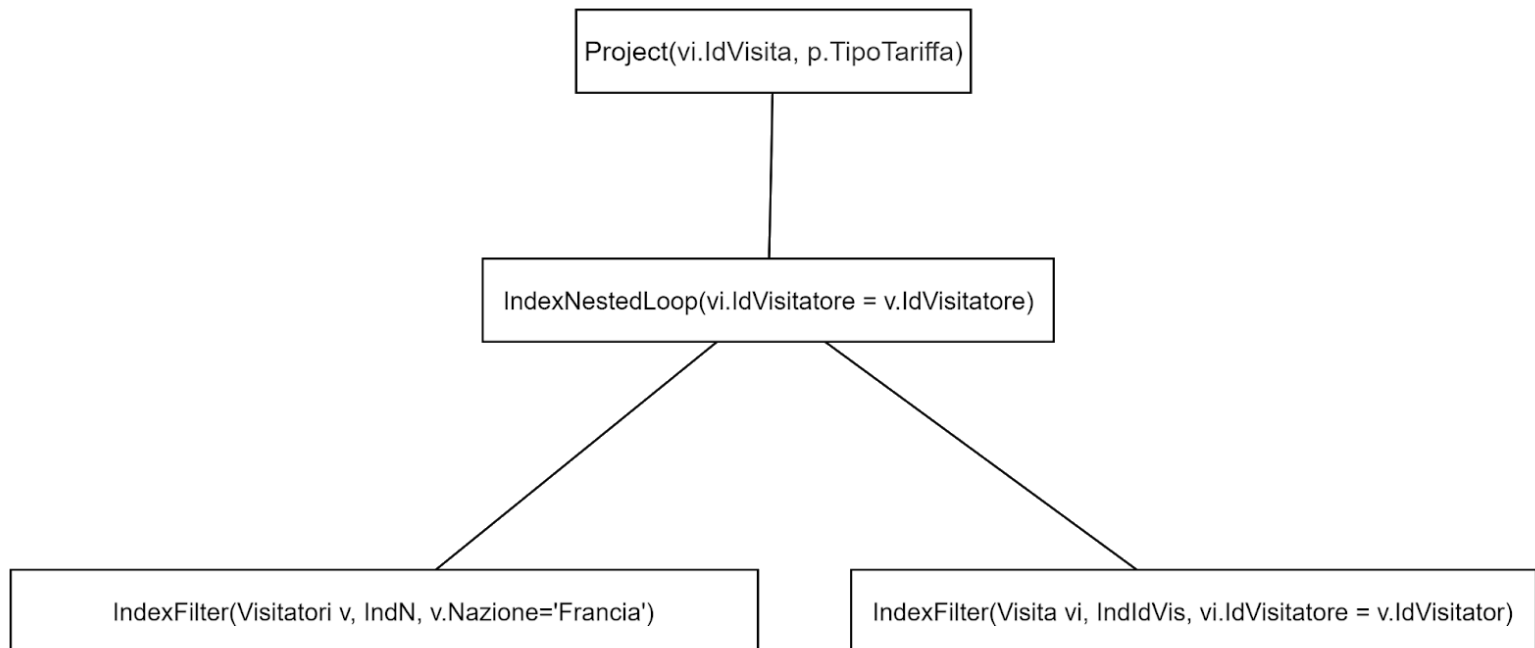
2.b. Piano di accesso fisico senza indici



2.c. Piano di accesso fisico senza indici



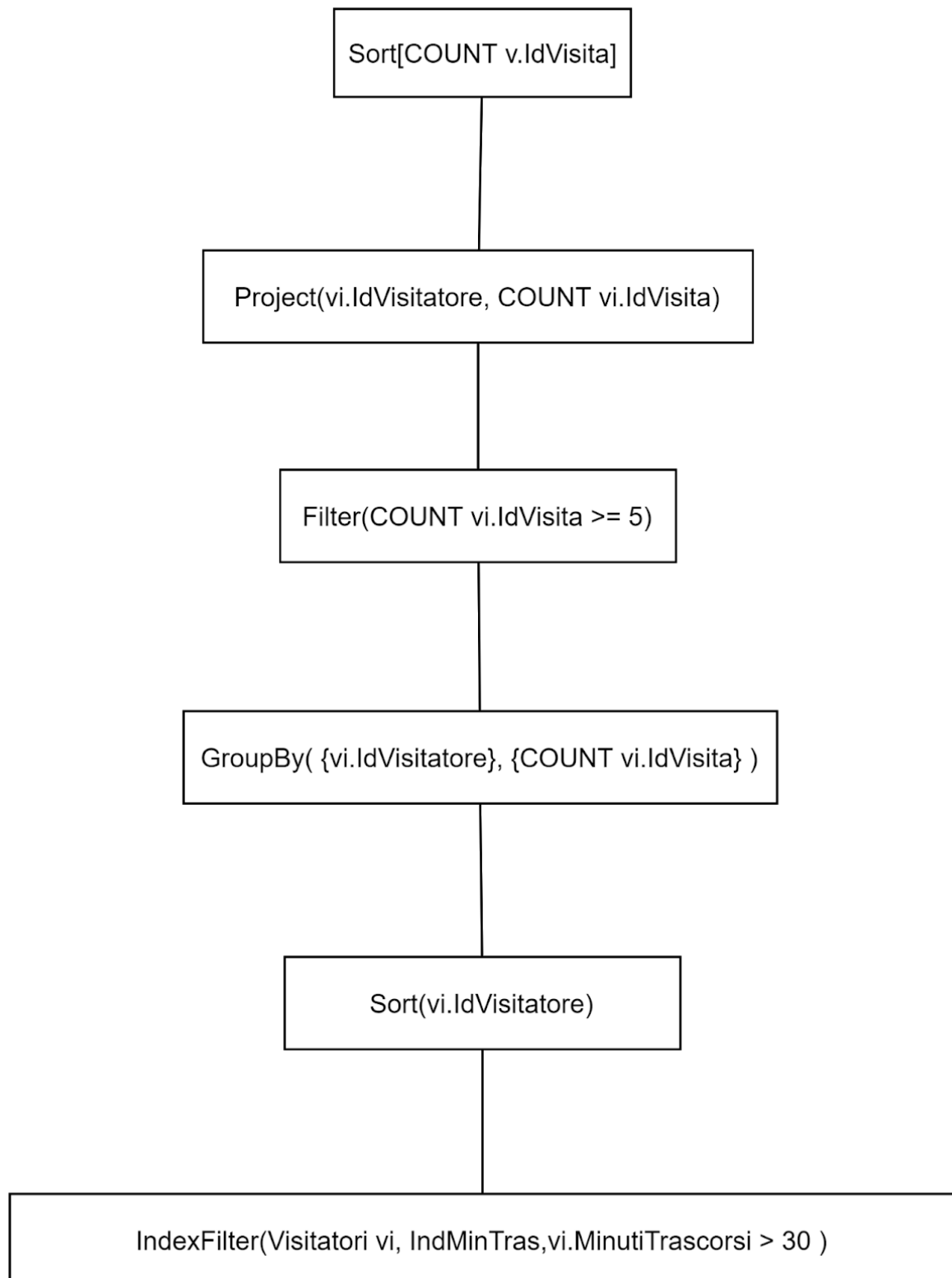
3.a. Piano di accesso fisico con indici



IndN = indice della tabella Visitatori sull'attributo Nazione

IndIdVis = indice della tabella Visita sull'attributo IdVisitatore

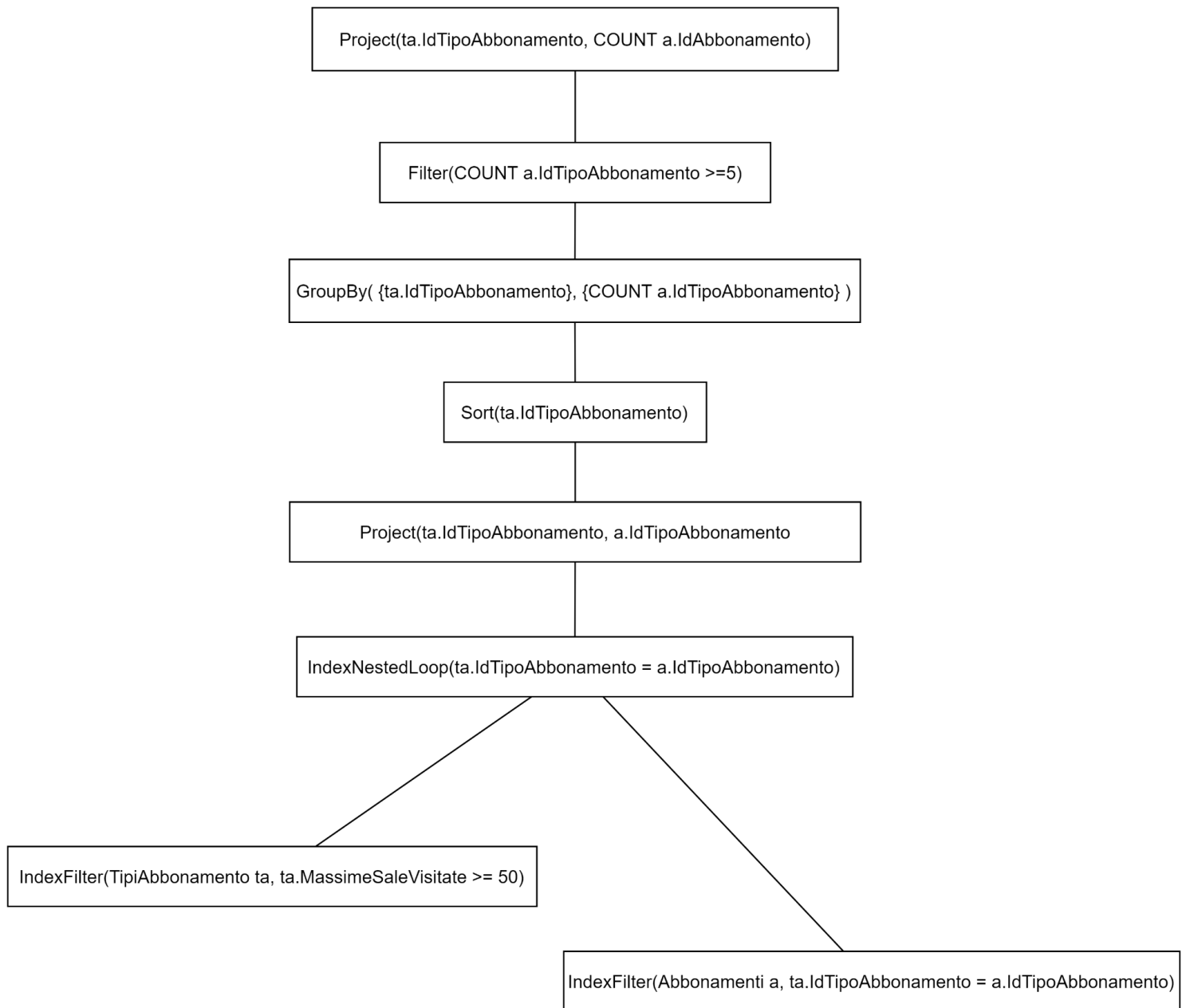
3.b. Piano di accesso fisico con indici



IndMinTras = indice della tabella Visite sull'attributo MinutiTrascorsi

[TORNA AL SOMMARIO](#)

3.c. Piano di accesso fisico con indici



IndTA = indice nella tabella Abbonamenti nell'attributo IdTipoAbbonamenti
IndMSV= indice nella tabella TipiAbbonamento nell'attributo MassimoSaleVisitabili