

SCI CLUB TORRE PENDENTE

Progetto Basi di dati 2022/2023

Appello Gennaio 2023

Consegnato il 10/01/2023

Riccardo Berni	616399
Alessandro Guerriero	596471
Elia Boccini	618967

Sci Club Torre Pendente

Lo sci club Torre Pendente accetta soci ordinari e soci atleti. Un socio è registrato se ha compilato correttamente il modulo e se ha pagato l'importo corretto della quota associativa, diversa tra ordinari e atleti. Dei soci interessano: nome, cognome, codice fiscale e il codice del pagamento della quota associativa. I soci atleti vengono anche tesserati alla FISI e devono obbligatoriamente caricare un certificato medico con la relativa data di rilascio. Il sistema deve mantenere il numero di tessera di ogni socio atleta. Il sistema salva anche i parchi sciistici, di cui interessano email e ubicazione. L'associazione permette di partecipare alle attività previste dal club, che sono:

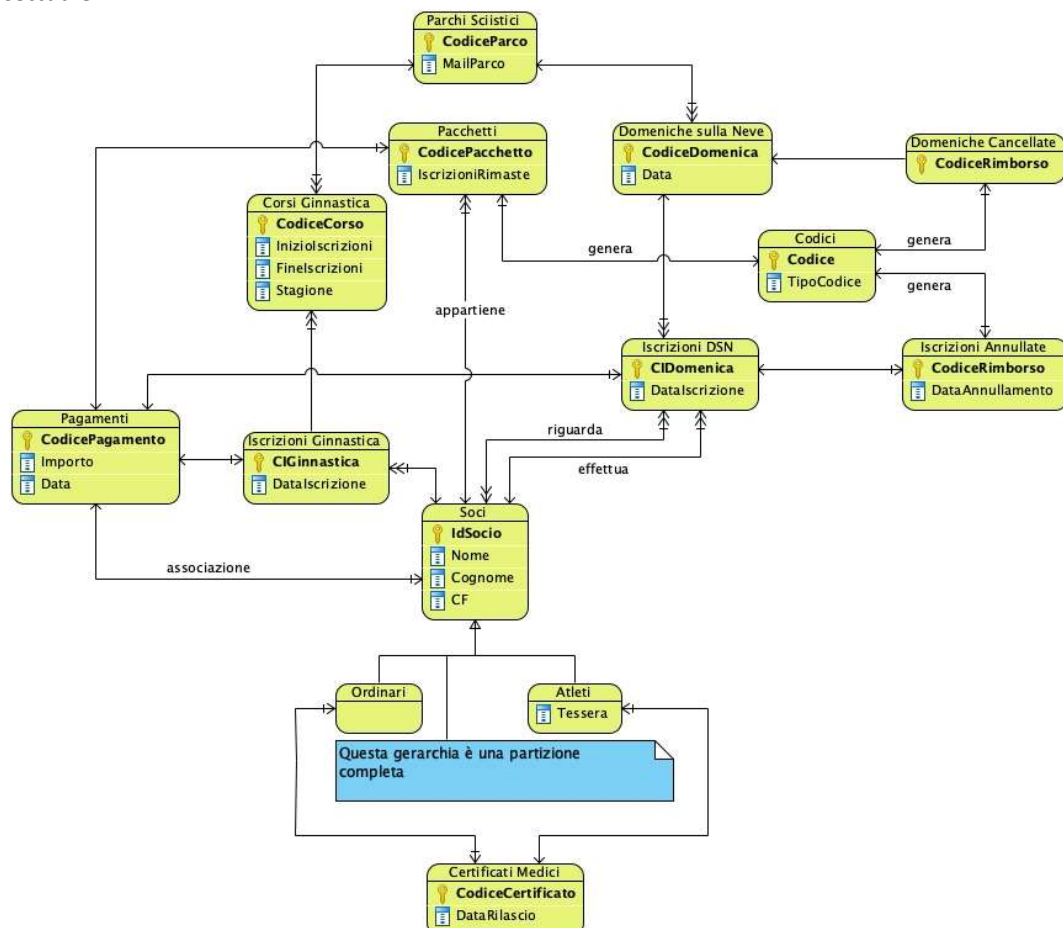
-Corso Ginnastica Presciistica, del quale interessa la data di inizio iscrizione e la data di fine iscrizione. Le iscrizioni al corso di ginnastica devono mantenere obbligatoriamente il codice del pagamento del bonifico e gli iscritti devono avere associato il certificato medico con la relativa data di rilascio.

-Domeniche sulla neve, di cui interessano data e parco sciistico. Le iscrizioni si effettuano compilando un modulo e effettuando un bonifico e devono tener conto della data di iscrizione, del socio iscritto (è possibile, con un unico modulo e bonifico, iscrivere più soci) e del codice del pagamento. Interessa inoltre se l'iscrizione sia stata fatta tramite codice o tramite pagamento.

Dei codici interessa il tipo di codice, che può essere:

- "pacchetto" che corrisponde a un pacchetto acquistato (associato a un socio e di cui si tiene conto delle iscrizioni rimanenti);
- "rimborso" che corrisponde al codice generato se una domenica viene cancellata o se un'iscrizione viene annullata entro una data e un'ora stabiliti dal club;
- "sconto", che corrisponde al codice generato se un'iscrizione viene annullata oltre la data limite e da diritto a uno sconto parziale.

Schema concettuale:



Schema relazionale:

Socio(IdSocio, Nome, Cognome, Email, CodFiscale, Tessera, CodPagamento*).

Pagamento(CodPagamento, Importo, Data).

CertificatoMedico(CodCertificato, DataRilascio, IdSocio*).

IscrizioneGinnastica(CIginnastica, Data, CodPagamento*, IdSocio*, CodCorso*).

CorsoGinnastica(CodCorso, IniziIscrizioni, FineIscrizioni, Stagione, CodParco*).

ParcoSciistico(CodParco, Email, Luogo).

IscrizioneDSN (CIIDomenica, DataIscrizione, IdSocio*, CodDomenica*, CodPagamento*, Codice).

IscrizioneAnnullata(CodiceRimborso*, CodIscrizione*, DataAnnullamento).

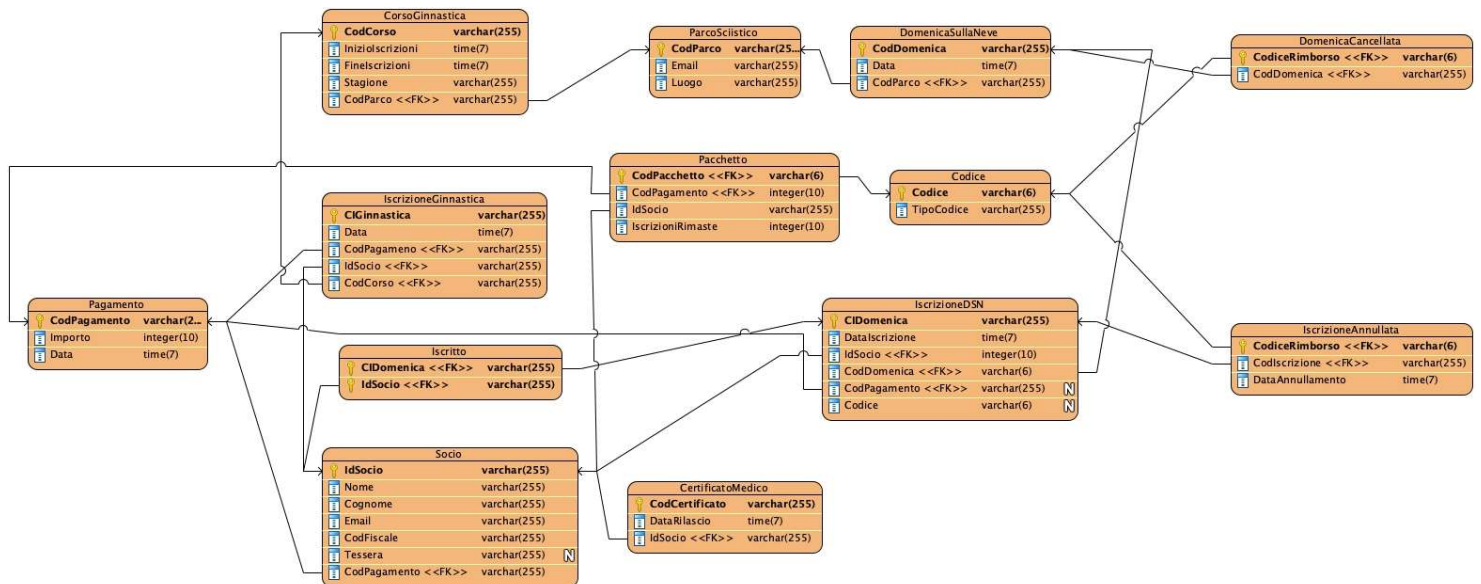
Iscritto(CIIDomenica*, IdSocio*).

DomenicaSullaNeve(CodDomenica, Data, CodParco*).

DomenicaCancellata(CodiceRimborso*, CodDomenica*).

Pacchetto(CodPacchetto*, CodPagamento*, IdSocio*, IscrizioniRimaste).

Codice(Codice, TipoCodice).



Normalità:

**le dipendenze funzionali legate alle chiavi non sono riportate*

Socio { **CodPagamento** -> IdSocio, Nome, Cognome, Email, CodFiscale, Tessera; **Email** -> IdSocio, Nome, Cognome, CodFiscale, Tessera, CodPagamento }

Pagamento ha solo dipendenze funzionali banali e quella legata alla chiave primaria

CertificatoMedico {**IdSocio** -> CodCertificato, DataRilascio}

IscrizioneGinnastica { **CodPagamento** -> ClGinnastica, Data, IdSocio, CodCorso }

CorsoGinnastica ha solo dipendenze funzionali banali e quella legata alla chiave primaria

ParcoSciistico {**Email** -> CodParco, Luogo}

IscrizioneDSN ha solo dipendenze funzionali banali e quella legata alla chiave primaria

IscrizioneAnnullata {**CodIscrizione** -> CodiceRimborso, DataAnnullamento}

Iscritto ha solo dipendenze funzionali banali e quella legata alla chiave primaria

DomenicaSullaNeve ha solo dipendenze funzionali banali e quella legata alla chiave primaria

DomenicaCancellata ha solo dipendenze funzionali banali e quella legata alla chiave primaria

Pacchetto { **CodPagamento** -> CodPacchetto, IdSocio, IscrizioniRimaste}

Codice ha solo dipendenze funzionali banali e quella legata alla chiave primaria

Quindi posso dire che tutte le relazioni soddisfano la forma normale di Boyce-Codd perché ogni relazione ha solamente dipendenze funzionali banali oppure dipendenze funzionali il cui insieme di attributi di sinistra è superchiave.

Vincoli:

**I vincoli sottolineati sono interrelazionali*

- Pagamento:
 - o Importo > 0.
- Socio:
 - o Se Tessera ≠ NULL allora idSocio deve essere presente nella tabella CertificadoMedico.
 - o Se Tessera ≠ NULL il pagamento associato deve avere importo = 50, altrimenti = 10 (quota stagionale).
- IscrizioneGinnastica:
 - o Data deve essere compresa tra iniziolscrizioni e Finelscrizioni del corso corrispondente a CodCorso.
- Corso Ginnastica:
 - o Finelscrizioni > Iniziolscrizioni.
 - o IdSocio deve essere presente in CertificadoMedico.
- Codice:
 - o Se Codice corrisponde a DomenicheCancellate il TipoCodice è "Rimborso", se corrisponde a Pacchetti è "Pacchetto". Se Codice corrisponde a IscrizioniAnnullate e DataAnnullamento è in ritardo TipoCodice è "Sconto", altrimenti "Rimborso".
- IscrizioneDSN:
 - o Codice deve essere presente nella tabella Codice.
 - o Se Codice è NULL CodPagamento non può essere NULL.
 - o Se Codice non è NULL CodPagamento è NULL se il tipo del codice corrispondente a Codice è "Pacchetto" o "Rimborso".
 - o Se Codice in Codice è di tipo "Pacchetto" questo non può essere utilizzato se IscrizioniRimaste = 0 in Pacchetto. Inoltre IdSocio deve essere uguale a IdSocio del pacchetto corrispondente a Codice.
- Pacchetto:
 - o IscrizioniRimaste è un contatore da 5 e viene decrementato quando CodPacchetto viene utilizzato.
- IscrizioneAnnullata:
 - o Se DataAnnullamento è entro la data limite il codice sarà di tipo "Rimborso", altrimenti di "Sconto".

Query:

1. Uso di proiezione, join e restrizione.

- *Voglio Nome, Cognome e Data di rilascio del certificato medico dei soci ordinari.*

```
SELECT S.Nome, S.Cognome, C.DataRilascio
FROM CertificatoMedico C JOIN Socio S ON S.IdSocio = C.IdSocio
WHERE Tessera IS NULL;
```

2. Uso di Group By con Having, Where e Sort. Si richiede inoltre che la relativa lista degli attributi dopo il costruito SELECT deve contenere almeno 3 elementi, due dei quali attributi della tabella.

- *Voglio sapere Nome, Cognome e cardinalità dei soci ordinari che hanno almeno un omonimo, ordinati per nome.*

```
SELECT Nome, Cognome, Count(*)
FROM Socio
WHERE Tessera IS NULL
GROUP BY Nome, Cognome
HAVING Count(*)>1
ORDER BY Nome;
```

3. Uso di Join, Group by con Having e Where.

- *Voglio conoscere i giorni in cui la somma degli importi dei pagamenti dei soci atleti sia almeno 200 e voglio anche mostrare la somma degli importi per giorno.*

```
SELECT P.DataPagamento, SUM(Importo)
FROM Pagamento P JOIN Socio S ON P.CodPagamento = S.CodPagamento
WHERE S.Tessera IS NOT NULL
GROUP BY P.DataPagamento
HAVING SUM(P.Importo) >= 200;
```

4. Uso di select annidata con quantificazione esistenziale.

- *Voglio conoscere IdSocio, Nome e Cognome dei soci che hanno acquistato almeno una domenica sulla neve all'Abetone.*

```
SELECT S.IdSocio, S.Nome, S.Cognome
FROM Socio S
WHERE S.IdSocio IN ( SELECT I.IdSocio
                     FROM IscrizioneDSN I, DomenicaSullaNeve D, ParcoSciistico P
                     WHERE I.CodDomenica = D.CodDomenica
                          AND D.CodParco = P.CodParco
                          AND P.Luogo = "Abetone");;
```

5. Uso di Select annidata con quantificazione esistenziale.

- *Voglio i CorsiPresciistica che hanno avuto come iscritti solo soci atleti.*

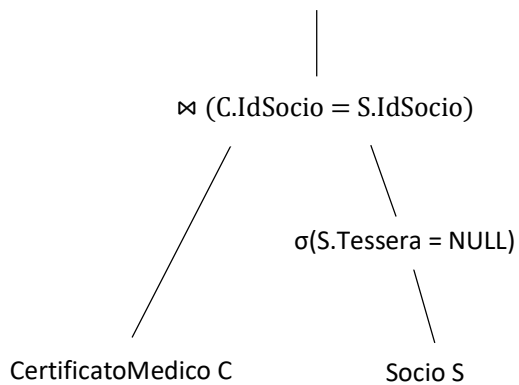
```
SELECT CodCorso
FROM CorsiPresciistica
WHERE CodCorso NOT IN ( SELECT CodCorso
                        FROM CorsiPresciistica C JOIN IscrizioneGinnastica I
                        ON C.CodCorso = I.CodCorso
                        JOIN Socio S ON S.IdSocio = C.IdSocio
                        WHERE Tessera IS NULL; );
```

6. Uso di subquery di confronto quantificato usando una subquery.
- *Voglio conoscere i giorni in cui la somma degli importi è maggiore della media della somma degli importi giornalieri, visualizzando la somma degli importi.*

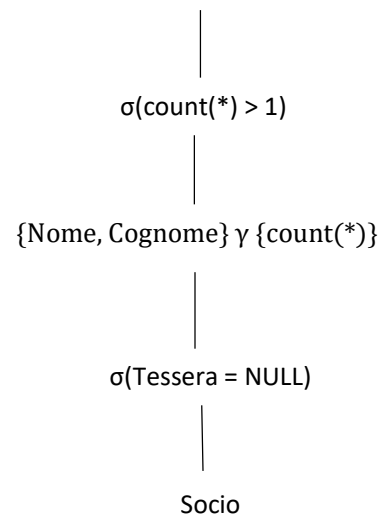
```
SELECT Data, SUM (Importo)
FROM Pagamento
GROUP BY Data
HAVING SUM(Importo) >      (SELECT AVG(SUM(Importo))
                             FROM Pagamento
                             GROUP BY Data);
```

Piani di accesso logici:

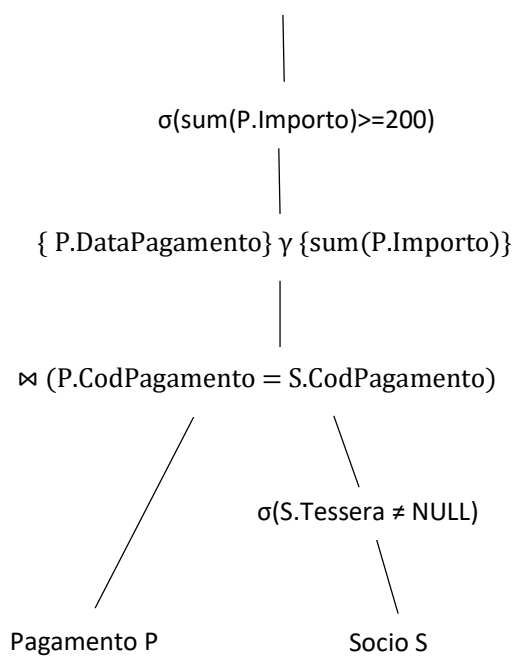
(1) $\pi (S.Nome, S.Cognome, C.DataRilascio)$



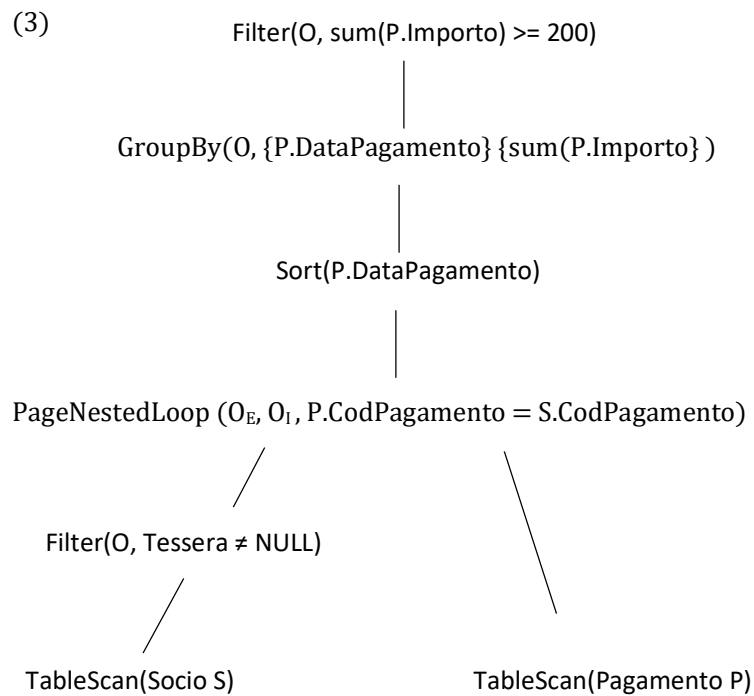
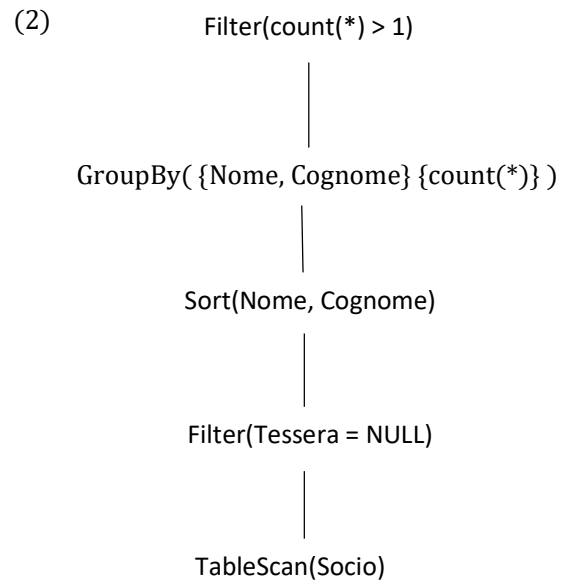
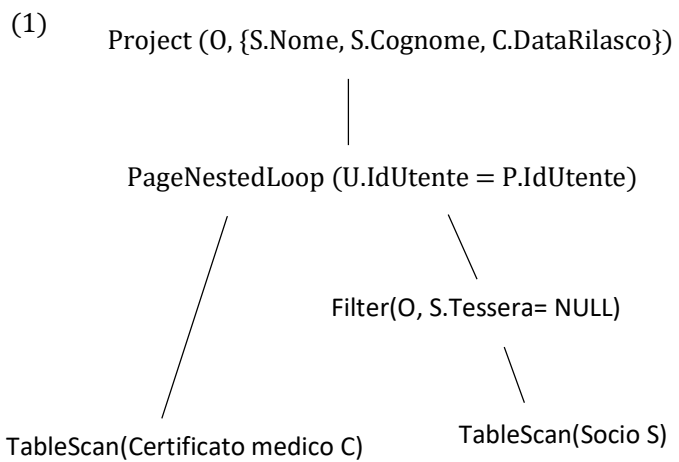
(2) $\pi (Nome, Cognome, count(*))$



(3) $\pi (P.DataPagamento, sum(P.Importo))$

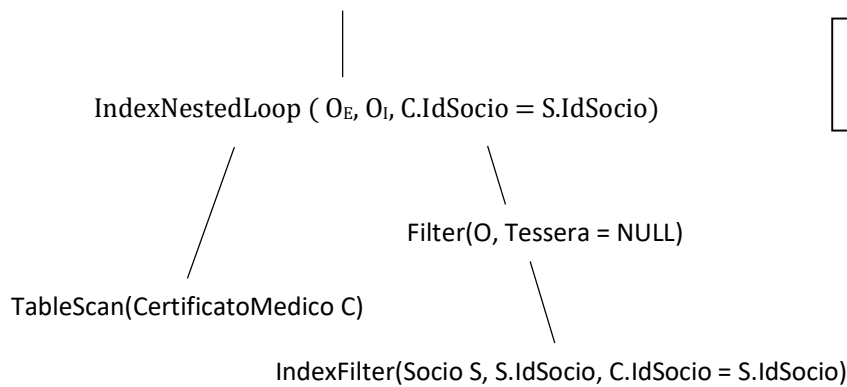


Piani di accesso fisici senza indice:



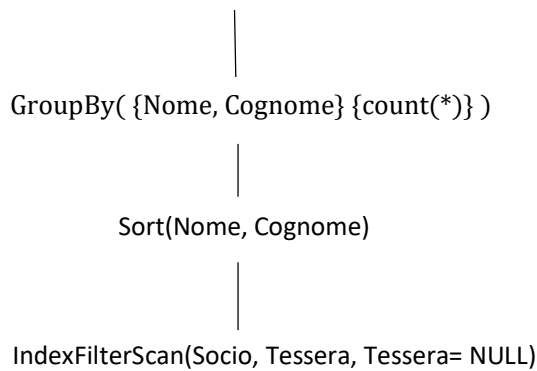
Piani di accesso fisici con indice:

(1) Project (O, {S.Nome, S.Cognome, C.DataRilascio})



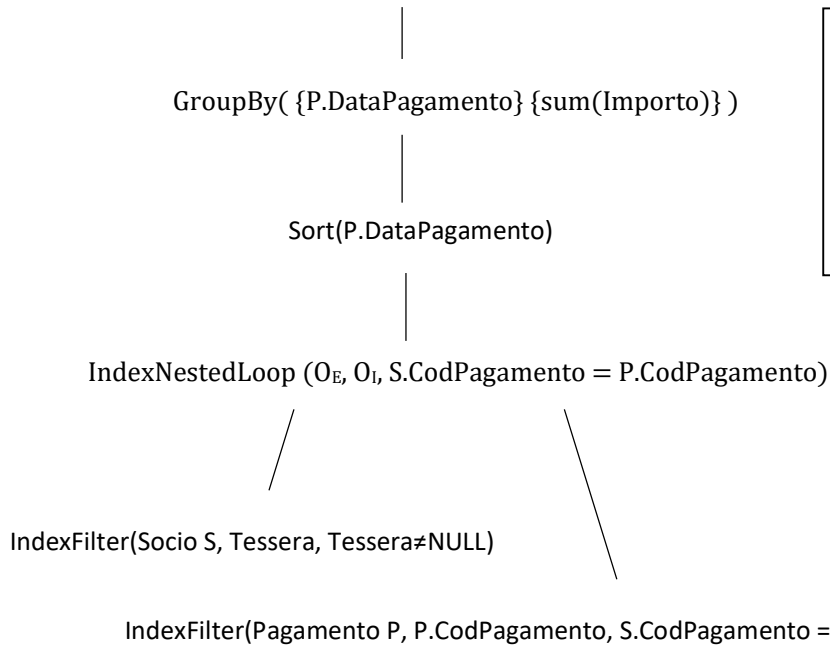
Creo l'indice su IdSocio in funzione dell' IndexNestedLoop.

(2) Filter(count(*) > 1)



Creo l'indice su Tessera perché trovo tutti i record con Tessera = NULL con un solo accesso all'albero.

(3) Filter(O, sum >= 200)



Creo l'indice su Tessera perché trovo tutti i record con Tessera ≠ NULL con un solo accesso all'albero.

L'indice di destra è creato su P.CodPagamento in funzione dell' IndexNestedLoop.