

Basi di Dati

A.A. 2019/2020

PROGETTO “Rockaway Beach!”

Simone Ianniciello – Matricola 581201 – Corso A

Emiliano Quaranta – Matricola 580667 – Corso B

## 1. Descrizione del dominio

### Cliente

La classe Cliente contiene il nome completo dei clienti, lo stato della blacklist e un numero di carta di credito.

Lo stato della blacklist è un intero con valori:

- 0 (def): nessuna infrazione effettuata o due prenotazioni consecutive andate a buon fine
- 1: prenotazione non andata a buon fine
- 2: prima prenotazione andata a buon fine dopo inserimento in blacklist
- 3: cliente bandito dopo la seconda infrazione

Se lo stato della blacklist di un cliente è pari a 3, esso non è più in grado di effettuare prenotazioni.

Il pagamento viene effettuato tramite carta di credito alle ore 16 del giorno precedente alla prenotazione giornaliera o alla data di inizio dell'abbonamento.

### Prenotazione

La classe Prenotazione mantiene uno storico delle prenotazioni effettuate negli ultimi 5 anni.

Ogni prenotazione è intestata a uno o più clienti.

Un cliente già associato a una prenotazione non è precluso dall'effettuarne un'altra.

### Abbonamento

La classe Abbonamento è una sottoclasse di Prenotazione e mantiene la data di inizio e la data di fine di un abbonamento.

### Giornaliero

La classe Giornaliero è una sottoclasse di Prenotazione e mantiene la data riferita alla prenotazione.

### PrezzoServizio

Della classe PrezzoServizio ci interessano, per ogni periodo compreso tra una data di inizio e una di fine, e ogni tipo di servizio, il prezzo fissato.

Si assume che non ci possano essere periodi di chiusura intermedi dall'inizio alla fine della stagione.

### Servizio

Nella classe Servizio viene specificato il tipo di servizio (ombrellone, gazebo, sdraio aggiuntive...).

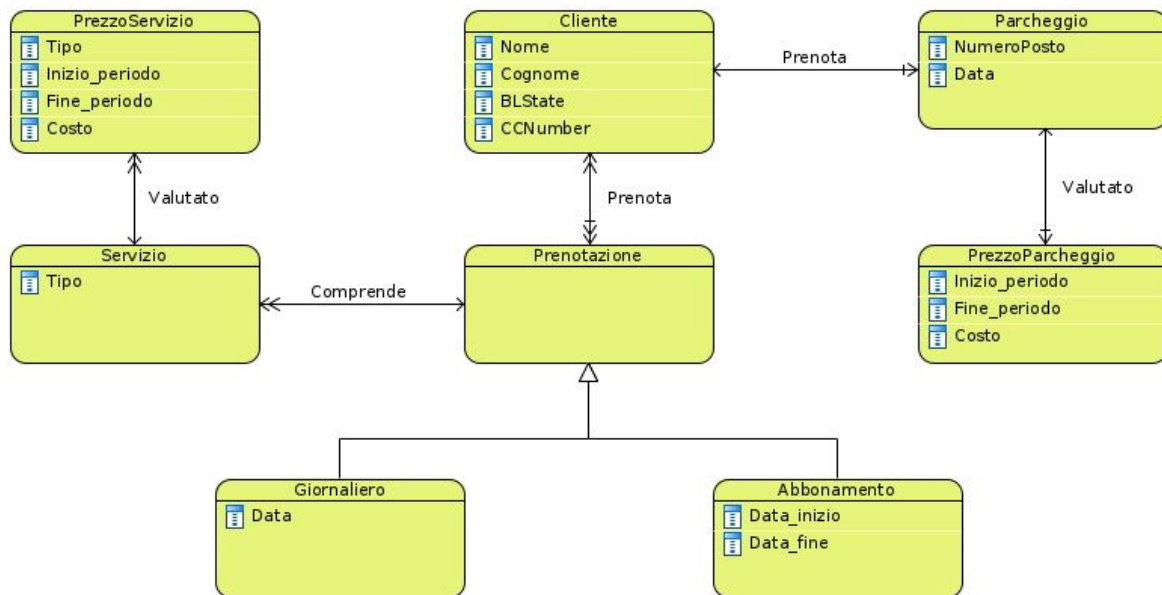
### Parcheggio

La classe Parcheggio mantiene uno storico delle prenotazioni dei parcheggi dell'ultima settimana e contiene il numero del posto assegnato e l'eventuale prezzo da pagare.

### PrezzoParcheggio

La classe PrezzoParcheggio definisce il costo del parcheggio, da tenere in conto solo nei weekend, differenziato per periodi.

## 2. Schema concettuale



### Vincoli intrarelazionali:

#### Cliente

BLState può essere nel range {0 = "Default"; 1 = "Ingresso in BL"; 2 = "Buona condotta"; 3 = "Bandito"}.

La lunghezza di CCNumber deve essere compresa tra 13 e 19 cifre.

#### Parcheggio

IdPst e Data formano la chiave della relazione.

#### PrezzoParcheggio

Inizio\_periodo < Fine\_periodo.

#### Abbonamento

Data\_inizio < Data\_fine.

#### Servizio

Tipo può essere nel range {0: "Ombrellone"; 1: "Gazebo"; 2: "Sdraio"; 3: "Lettino"; 4: "Sedia";

5: "Cabina senza doccia"; 6: "Cabina con doccia"}.

#### PrezzoServizio

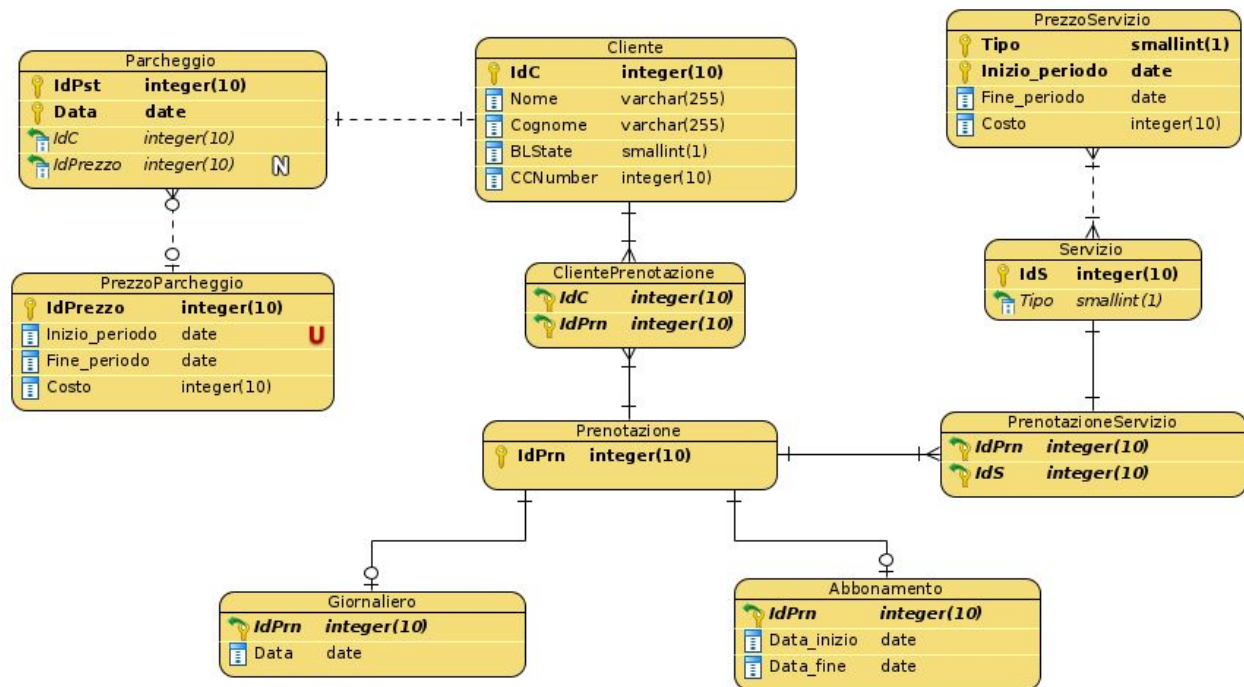
Inizio\_periodo e Tipo formano una chiave.

Inizio\_periodo < Fine\_periodo.

### Vincoli interrelazionali:

- Il numero di servizi richiesti in una prenotazione deve essere  $\geq 3$  (ombrellone/gazebo + due sdraio/lettini inclusi e eventuali servizi extra).
- Se il valore di BLState di un cliente è pari a 3, il cliente non deve essere più in grado effettuare una prenotazione.
- Data\_inizio associato a Abbonamento deve essere  $\geq$  al minimo Inizio\_periodo nella classe PrezzoServizio.
- Data associato a Giornaliero deve essere  $\geq$  al minimo Inizio\_periodo nella classe PrezzoServizio.
- Data\_fine associato a Abbonamento deve essere  $\leq$  al massimo Fine\_periodo nella classe PrezzoServizio.
- Data associato a Giornaliero deve essere  $\leq$  al massimo Fine\_periodo nella classe PrezzoServizio.

### 3. Schema logico relazionale



### Schema logico testuale

Cliente(IdC, Nome, Cognome, BLState, CCNumber)

ClientePrenotazione(IdC\*, IdPrn\*)

Prenotazione(IdPrn)

Abbonamento(IdPrn\*, Data\_inizio, Data\_fine)

Giornaliero(IdPrn\*, Data)

PrenotazioneServizio(IdPrn\*, IdS\*)

Servizio(IdS, Tipo\*)

PrezzoServizio(Tipo, Inizio\_periodo, Fine\_periodo, Costo)

Parcheggio(IdPst, Data, IdC\*, IdPrezzo\*)

PrezzoParcheggio(IdPrezzo, Inizio\_periodo, Fine\_periodo, Costo)

## Dipendenze funzionali:

### Cliente

IdC -> Nome, Cognome, BLState, CCNumber: Rispetta la BCNF perchè IdC è chiave.

### Abbonamento

IdPrn -> Data\_inizio, Data\_fine: Rispetta la BCNF perchè IdPrn è chiave.

### Giornaliero

IdPrn -> Data: Rispetta la BCNF perchè IdPrn è chiave.

### Servizio

IdS -> Tipo: Rispetta la BCNF perchè IdS è chiave.

### PrezzoServizio

Tipo, Inizio\_periodo -> Fine\_periodo, Costo: Rispetta la BCNF perchè {Tipo, Inizio\_periodo} è chiave.

### Parcheggio

IdPst, Data -> IdC, IdPrezzo: Rispetta la BCNF perchè {IdPst, Data} è chiave.

IdC, Data -> IdPst, IdPrezzo: Rispetta la BCNF perchè {IdC, Data} è chiave.

### PrezzoParcheggio

IdPrezzo -> Inizio\_periodo, Fine\_periodo, Costo: Rispetta la BCNF perchè IdPrezzo è chiave.

Inizio\_periodo -> IdPrezzo: Rispetta la BCNF perchè Inizio\_periodo è chiave.

## 4. Interrogazioni in SQL

a. Lista dei clienti associati alla prenotazione numero 123

```
SELECT Nome, Cognome, BLState
FROM Cliente
    NATURAL JOIN ClientePrenotazione
WHERE IdPrn = 123
```

b. Tabella PrezzoServizio divisa per Inizio\_periodo con Inizio\_periodo superiore al 01/06 e per le quali la somma dei prezzi di ogni servizio è maggiore di 40

```
SELECT *
FROM PrezzoServizio
WHERE Inizio_periodo > 01/06
GROUP BY Inizio_periodo
HAVING SUM(Costo) > 40
SORT BY Tipo
```

c. Lista di clienti con BLState pari a 1 che hanno effettuato più di 10 prenotazioni

```
SELECT IdC
FROM Cliente
    NATURAL JOIN ClientePrenotazione
WHERE BLState = 1
GROUP BY IdPrn
HAVING COUNT(IdPrn) > 10
```

d. Lista di abbonamenti che hanno richiesto almeno una sedia

```
SELECT a.IdPrn
FROM Abbonamento a
WHERE EXISTS (
    SELECT *
    FROM PrenotazioneServizio ps
        NATURAL JOIN Servizio s
    WHERE a.IdPrn = ps.IdPrn AND s.Tipo = 4
)
```

e. Lista di utenti che non hanno mai richiesto un gazebo in una prenotazione

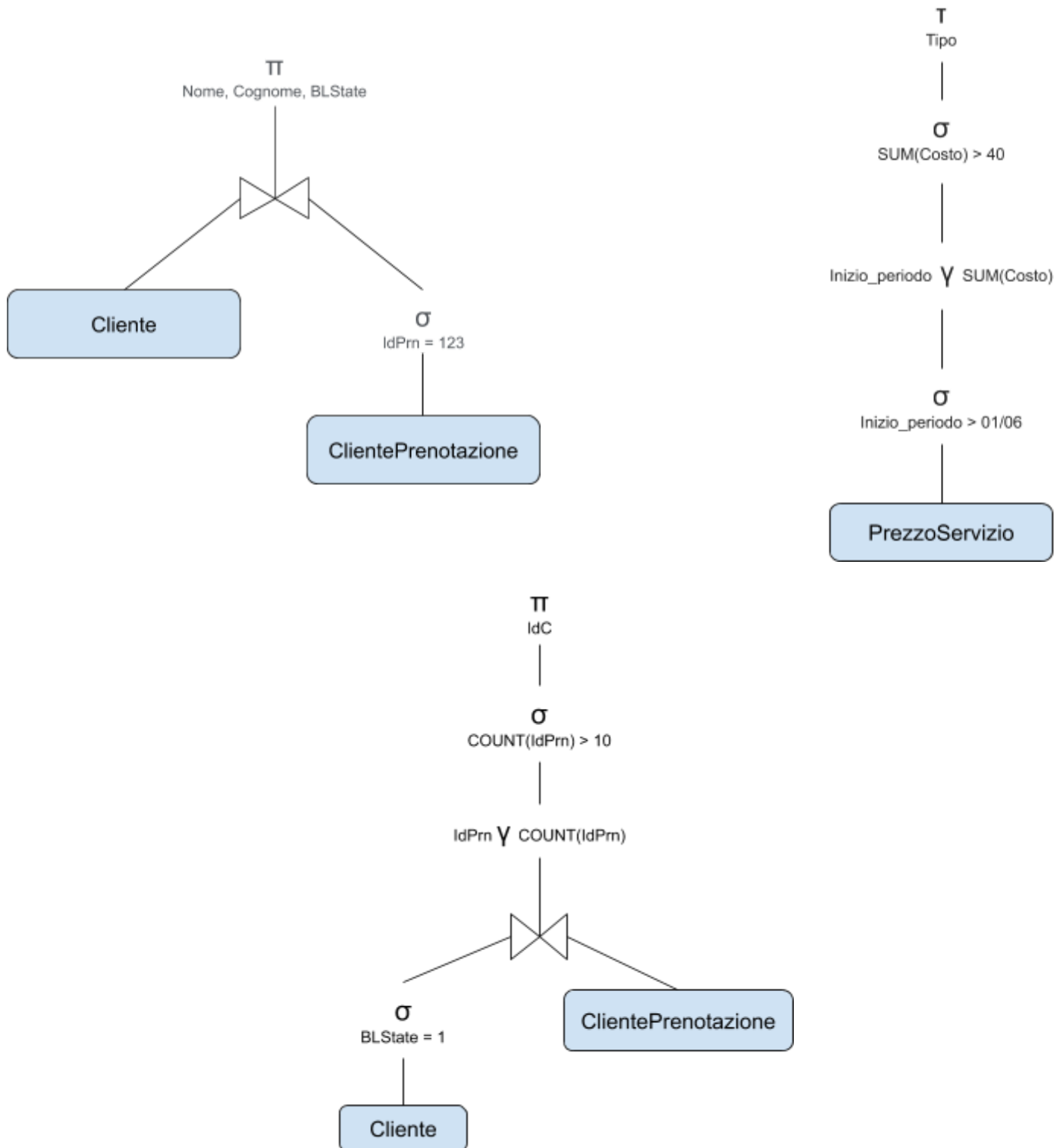
```
SELECT u.IdC
FROM Utente u
WHERE NOT EXISTS (
  SELECT *
  FROM ClientePrenotazione cp
    NATURAL JOIN Prenotazione p
    NATURAL JOIN PrenotazioneServizio ps
    NATURAL JOIN Servizio s
  WHERE u.IdC = cp.IdC AND s.Tipo != 2
)
```

f. Lista di abbonamenti che hanno richiesto 5 o più servizi

```
SELECT a.IdPrn
FROM Abbonamento a
WHERE 5 <= (
  SELECT COUNT(*)
  FROM PrenotazioneServizio ps
  WHERE a.IdPrn = ps.IdPrn
)
```

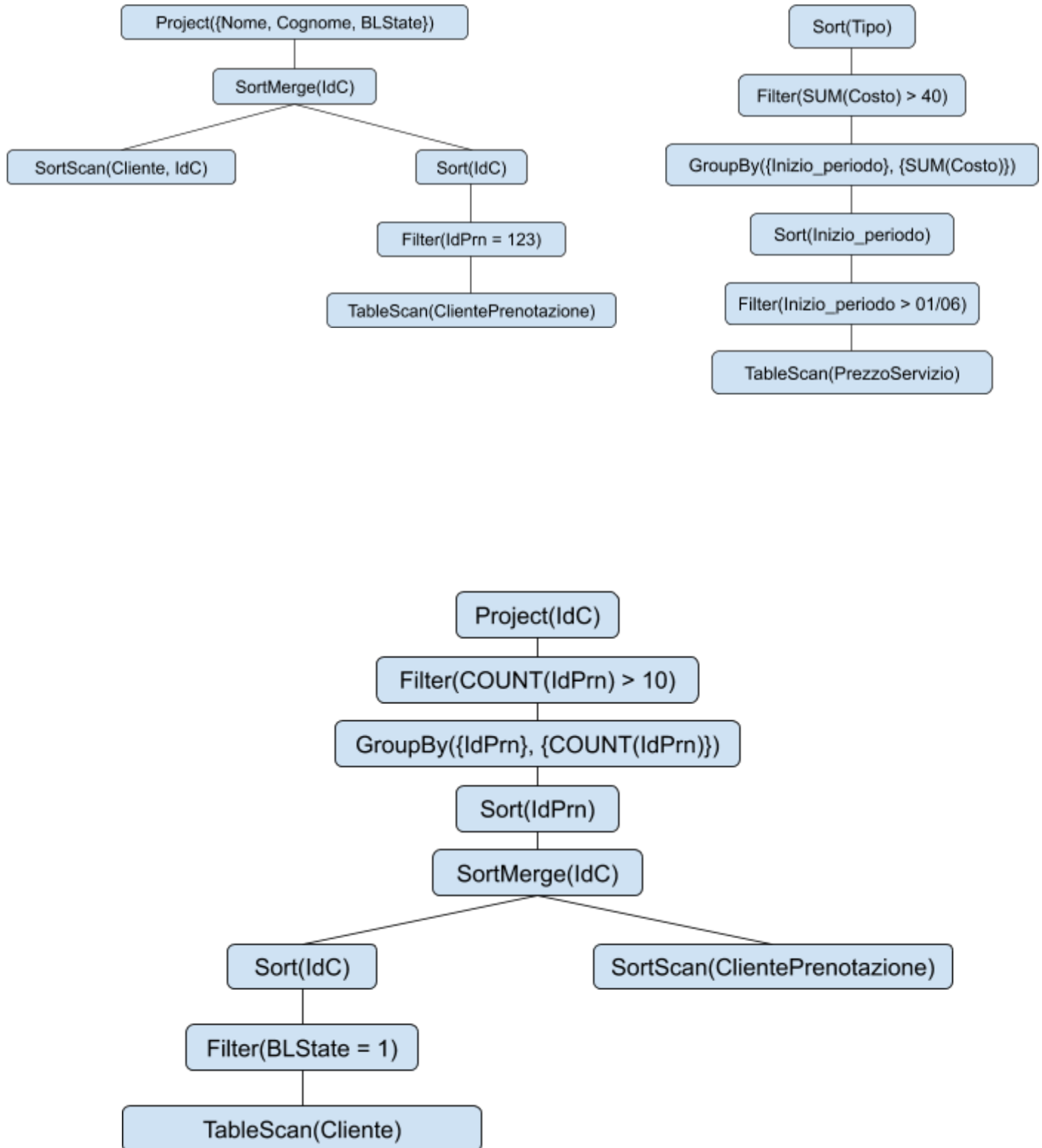
## 5. Piani di accesso

### Piani di accesso logico





## Piani di accesso fisico (senza indici)



## Piani di accesso fisico (con indici)

