

*Corso di laurea in Informatica
Corso di Basi di Dati - Pari
Dipartimento di Informatica Università di Pisa*

“Un giorno al museo”

*a.a 2020-21
Docente Giorgio Ghelli*



Giacomo Aru	597700	g.aru@studenti.unipi.it
Dario Ayrton Corveddu	597748	d.corveddu@studenti.unipi.it
Simone Marzeddu	597134	s.marzeddu@studenti.unipi.it
Jacopo Raffi	598092	j.raffi@studenti.unipi.it



Data di Consegna: 04/06/2021

1. Descrizione del dominio

“Ispirandosi alla descrizione fornita, lo studente dà una descrizione precisa del dominio del discorso”

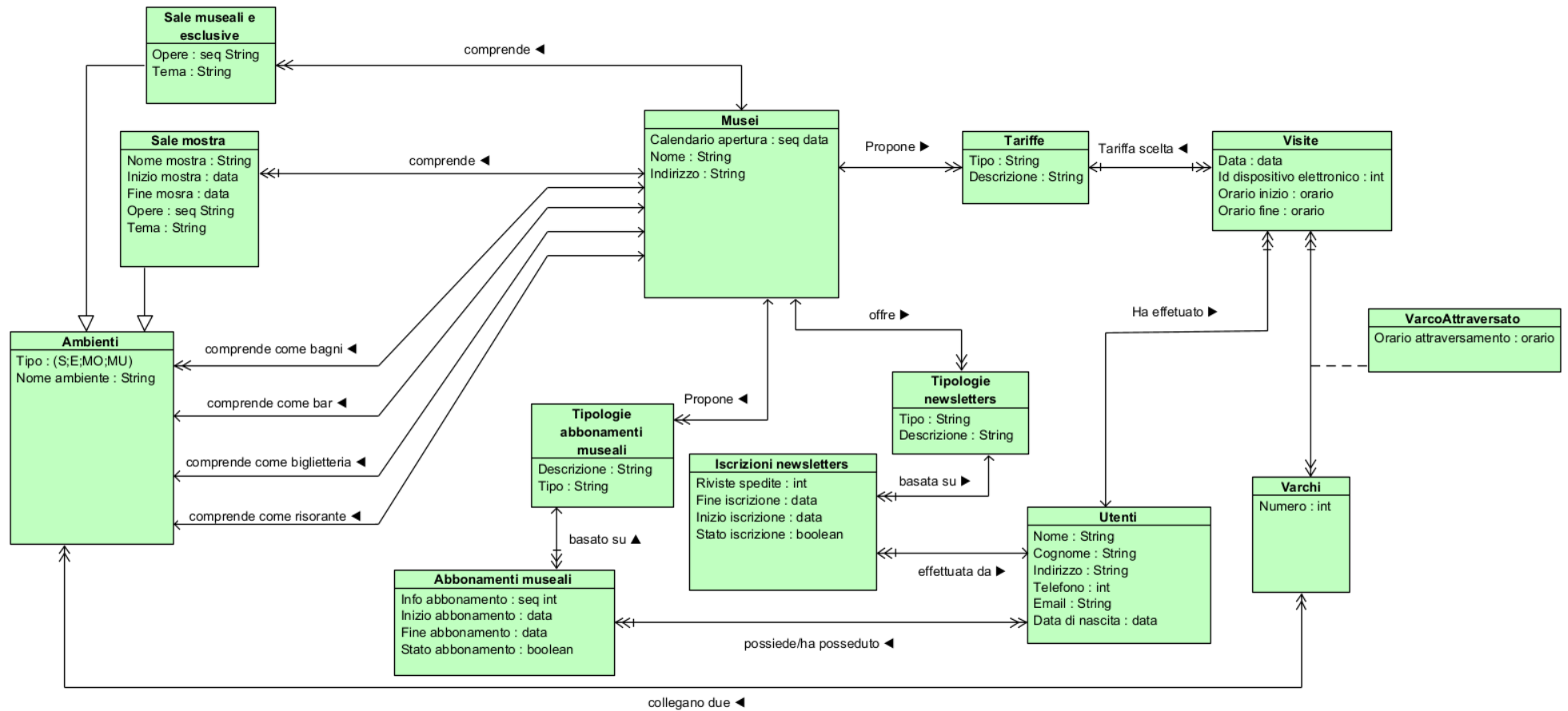
Un **ambiente** è una sala che si può distinguere in sala mostra, sala museale, sala esclusiva o sala servizio in base all'utilizzo a cui è adibita. Un **varco** è un collegamento tra due ambienti all'interno del museo e può collegare tra loro sale di qualsiasi tipo. Un **museo** è un insieme di ambienti con almeno una sala museale, un numero variabile di stanze tra sale mostra e sale esclusive, un bar, un ristorante, una biglietteria e almeno un bagno. Esso offre diverse tariffe, due **tipologie di abbonamenti museali** (semplice e speciale) e più **tipologie di newsletters**.

Un **utente** è un visitatore che usufruisce di uno dei due servizi del museo (visita e newsletter). Una **visita** è un itinerario all'interno del museo che prevede l'assegnazione di un dispositivo elettronico di supporto e, se non si ha un abbonamento attivo, la scelta di una tariffa. Una **tariffa** è una modalità di calcolo del costo della visita per utenti non abbonati, e ne esistono due tipologie differenti: bianca e verde. Un utente può effettuare un **abbonamento museale** (singolo o multiplo) per un diverso calcolo del costo di una o più visite future. Un utente può anche effettuare una o più **iscrizioni alle newsletters** così da ricevere la rivista periodicamente.

Nota: le parole in **grassetto** rappresentano le classi del dominio, le parole sottolineate sono sottoclassi.

2. Schema concettuale

“Lo studente produce uno schema concettuale a oggetti in formato grafico”



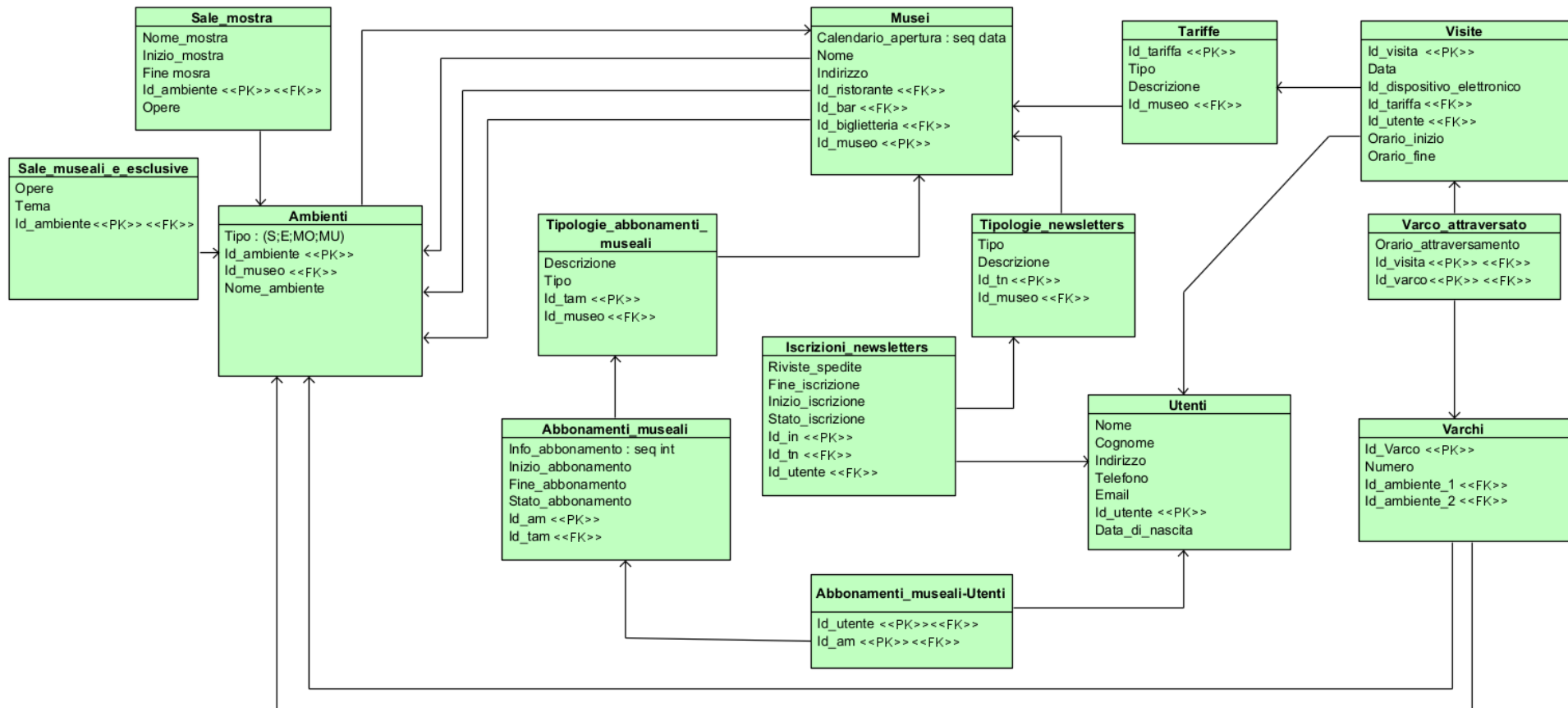
Vincoli interrelazioni e intrarelazionali individuati:

1. un utente possiede un abbonamento attivo and effettua una visita al museo se e solo se la visita non ha nessuna tariffa associata
2. tutti i bagni, il ristorante, il bar e la biglietteria presenti nel museo sono ambienti di tipo S
3. un utente ha solo un abbonamento con stato TRUE
4. un varco connette due ambienti diversi
5. data una visita, il varco associato con l'orario di attraversamento minore e il varco associato con l'orario di attraversamento maggiore devono affacciarsi sulla biglietteria
6. per tutte le coppie (orario inizio, orario fine) deve valere: $\text{orario inizio} < \text{orario fine}$
7. per tutte le coppie (data inizio, data fine) deve valere: $\text{data inizio} \leq \text{data fine}$

3.Schema logico relazionale

“Lo studente produce uno schema logico relazionale che corrisponde allo schema del punto 2.”

Schema grafico



Schema testuale

Museo(Id_museo, Id_ristorante*, Id_biglietteria*, Id_bar*, Nome, Indirizzo, Calendario_apertura)
Tariffe(Id_tariffa, Id_museo*, Tipo, Descrizione)
Visite(Id_visita, Id_tariffa*, Id_utente*, Id_dispositivo_elettronico, Data, Orario_inizio, Orario_fine)
Varco_attraversato(Id_visita*, Id_varco*, Orario_attraversamento)
Varchi(Id_varco, Id_ambiente_1*, Id_ambiente_2*, Numero)
Ambienti(Id_ambiente, Id_museo*, Nome_ambiente, Tipo)
Sale_museali_e_esclusive(Id_ambiente*, Opere, Tema)
Sale_mostra(Id_ambiente*, Nome_mostra, Inizio_mostra, Fine_mostra, Opere)
Tipologie_abbonamenti_museali(Id_tam, Id_museo*, Tipo, Descrizione)
Abbonamenti_museali(Id_am, Id_tam*, Info_abbonamento, Inizio_abbonamento, Fine_abbonamento, Stato_abbonamento)
Abbonamenti_museali-Utenti(Id_utente*, Id_am*)
Utenti(Id_utente, Nome, Cognome, Indirizzo, Telefono, Email, Data_di_nascita)
Iscrizioni_newsletters(Id_in, id_tn*, Id_utente*, Riviste_spedite, Inizio_iscrizione, Fine_iscrizione, Stato_iscrizione)
Tipologie_newsletters(Id_tn, Id_museo*, Tipo, Descrizione)

Dipendenze funzionali

BNF(si/no)

Museo:

Id_museo	⇒	id_ristorante, Id_biglietteria, Id_bar, Nome, Indirizzo, Calendario_apertura	si
Nome, Indirizzo	⇒	Id_museo	si

Tariffe:

Id_tariffe	⇒	Id_museo, Tipo, Descrizione	si
------------	---	-----------------------------	----

Visite:

Id_visita	⇒	Id_tariffa, Id_utente, Id_dispositivo_elettronico, Data, Orario_inizio, Orario_Fine	si
Id_utente, Orario_inizio, Data	⇒	Id_visita	si
Id_utente, Orario_fine, Data	⇒	Orario_inizio	si
Orario_inizio, Data, Id_dispositivo_elettronico	⇒	Orario_fine, id_utente	si

Varco_attraversato:

Id_visita	⇒	Id_varco, Orario_attraversamento	si
-----------	---	----------------------------------	----

Varchi:

Id_varco	⇒	Id_ambiente_1, Id_ambiente_2, Numero	si
----------	---	--------------------------------------	----

Id_ambiente_1, Id_ambiente_2	⇒	Id_varco	si
------------------------------	---	----------	----

Numero	⇒	Id_varco	si
--------	---	----------	----

Ambienti:

Id_ambiente	⇒	Id_museo, Nome_ambiente, Tipo	si
-------------	---	-------------------------------	----

Nome_ambiente	⇒	Id_ambiente	si
---------------	---	-------------	----

Sale_museali_e_esclusive:

Id_ambiente	⇒	Opere, Tema	si
-------------	---	-------------	----

Opere	⇒	Id_ambiente	si
-------	---	-------------	----

Sale_mostra:

Id_ambiente	⇒	Nome_mostra, Inizio_mostra, Fine_mostra, Opere	si
-------------	---	--	----

Opere	⇒	Id_ambiente	si
-------	---	-------------	----

Tipologie_abbonamenti_museali:

Id_tam	⇒	Id_museo, Tipo, Descrizione	si
--------	---	-----------------------------	----

Tipo	⇒	Id_tam	si
------	---	--------	----

Abbonamenti_museali:

Id_am	⇒	Id_tam, Info_abbonamento, Inizio_abbonamento, Fine_abbonamento, Stato_abbonamento	si
-------	---	---	----

Utenti:

Id_utente	⇒	Nome, Cognome, Indirizzo, Telefono, Email, Data_di_nascita	si
-----------	---	--	----

Iscrizioni_newsletters:

Id_in	⇒	id_tn, Id_utente, Riviste_spedite, Inizio_iscrizione, Fine_iscrizione, Stato_iscrizione	si
-------	---	---	----

id_tn, Id_utente, Inizio_iscrizione	⇒	Id_in	si
-------------------------------------	---	-------	----

id_tn, Inizio_iscrizione, Fine_iscrizione	⇒	Riviste_spedite	no
---	---	-----------------	----

Tipologie_newsletters

Id_tn	⇒	Id_museo, Tipo, Descrizione	si
-------	---	-----------------------------	----

Tipo	⇒	Id_tn	si
------	---	-------	----

4. Interrogazioni in SQL

“Lo studente definisce un elenco di almeno 6 operazioni relative al dominio specificato al punto 1”

a. Uso di proiezione, join e restrizione;

Nome, cognome, inizio abbonamento e info abbonamento di tutti gli utenti che hanno un abbonamento al museo attivo.

```
SELECT u.Nome, u.Cognome, a.Inizio_abbonamento, a.Info_abbonamento  
FROM Utenti u, Abbonamenti_museali a, Abbonamenti_museali-Utenti au  
WHERE u.Id_utente = au.Id_utente AND a.id_am = au.Id_am AND a.Stato_abbonamento = TRUE
```

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

Nome, cognome, numero di riviste totali spedite e numero di iscrizioni totali, per ogni utente che ha ricevuto almeno 3 riviste e che si è iscritto alla newsletter nel 2020, in ordine crescente di numero di riviste.

```
SELECT u.Nome, u.Cognome, SUM(i.Riviste_spedite), COUNT(*)  
FROM Iscrizioni_newsletters i, Utenti u  
WHERE i.Inizio_Iscrizione >= 01/01/2020 AND u.Id_utente = i.Id_utente  
GROUP BY i.Id_utente, u.Nome, u.Cognome  
HAVING SUM(i.Riviste_spedite) >= 3  
ORDER BY SUM(i.Riviste_spedite)
```


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

Nome, cognome, data di nascita e numero di visite totali con una tariffa selezionata di tutti gli utenti nati dopo il 31/12/2004.

```
SELECT u.Nome, u.Cognome, u.Data_di_nascita, COUNT(*)  
FROM Utenti u  
JOIN Visite v ON u.Id_utente = v.Id_utente  
WHERE v.Id_tariffa <> NULL  
GROUP BY i.Id_utente, u.Nome, u.Cognome, u.Data_di_nascita  
HAVING v.Data_di_nascita > 31/12/2004
```

d. uso di select annidata con quantificazione esistenziale;

Nome ambiente, tipo ambiente di tutti gli ambienti che hanno almeno un varco collegato alla biglietteria

```
SELECT a.Nome_ambiente, a.Tipo  
FROM Ambienti a1  
WHERE EXISTS( SELECT *  
                FROM Ambienti a2, Varchi v  
                WHERE a2.Nome_ambiente = "Biglietteria" AND (v.Id_ambiente_1 = a2.Id_ambiente OR  
                    v.Id_ambiente_2 = a2.Id_ambiente ) AND (a1.Id_ambiente = v.Id_ambiente_1 OR  
                    a1.Id_ambiente = v.Id_ambiente_2) AND a1.Id_ambiente <> a2.Id_ambiente)
```

e. uso di select annidata con quantificazione universale;

Nome ambiente, tipo ambiente di tutti gli ambienti che non hanno nessun varco collegato a ambienti di servizio

```
SELECT a.Nome_ambiente, a.Tipo
FROM Ambienti a1
WHERE NOT EXISTS( SELECT *
                   FROM Ambienti a2, Varchi v
                   WHERE a2.Tipo = S AND (v.Id_ambiente_1 = a2.Id_ambiente OR v.Id_ambiente_2 = a2.Id_ambiente)
                   AND (a1.Id_ambiente = v.Id_ambiente_1 OR a1.Id_ambiente = v.Id_ambiente_2) AND
                   a1.Id_ambiente <> a2.Id_ambiente)
```

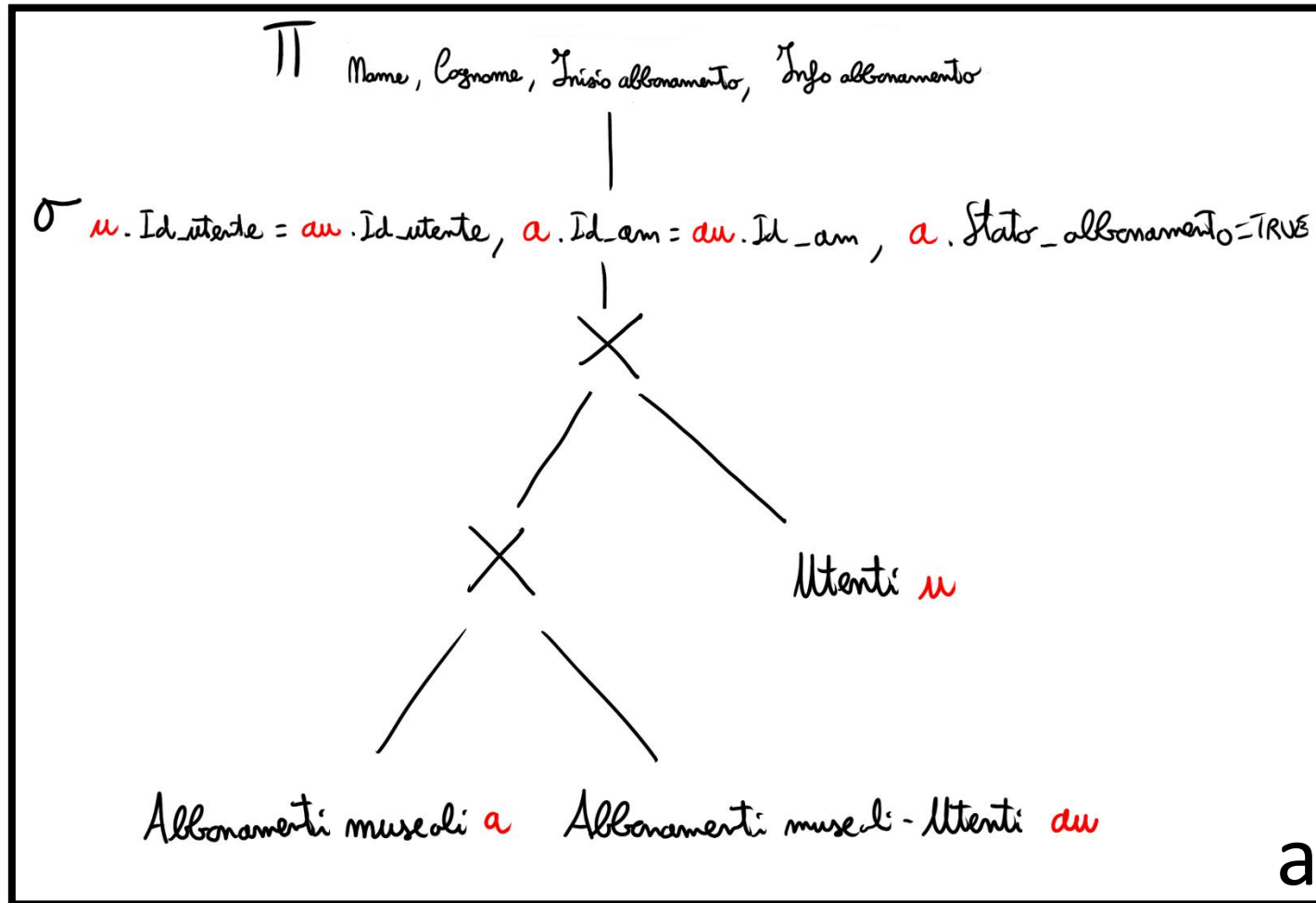
f. uso di subquery di confronto quantificato usando una subquery.

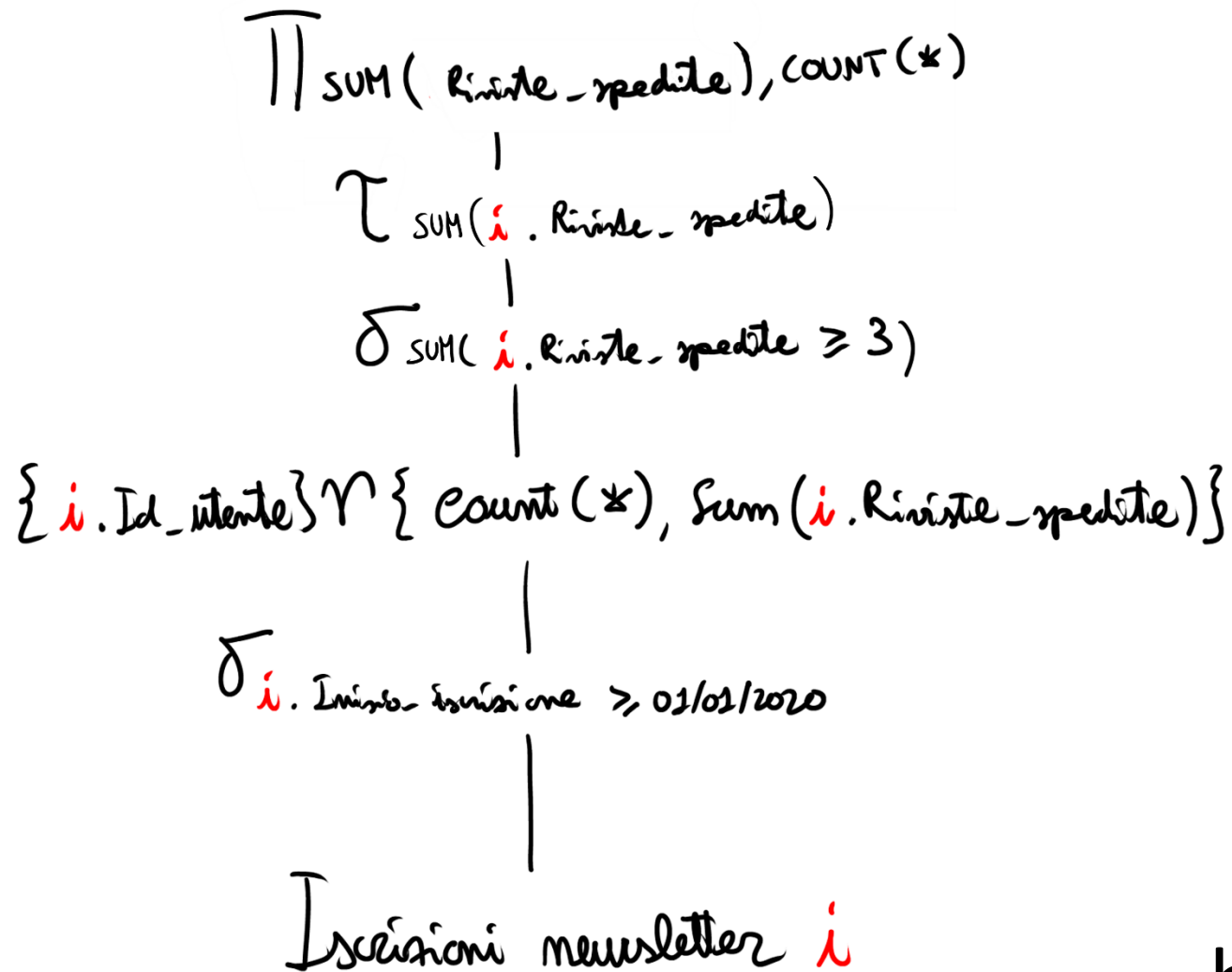
Tipo e descrizione di ogni tipologia di newsletters che ha un numero di iscritti maggiore al numero di abbonamenti speciali al museo

```
SELECT tn.Tipo, tn.Descrizione
FROM Tipologia_newsletters tn, Iscrizioni_newsletters in
WHERE in.Id_tn = tn.id_tn
GROUP BY tn.Id_tn, tn.Tipo, tn.Descrizione
HAVING COUNT(*) > ( SELECT COUNT(*)
                   FROM Tipologie_abbonamento_museale tam, Abbonamenti_museali am
                   WHERE tam.Id_tam = am.Id_tam AND tam.Tipo = "Speciale")
```

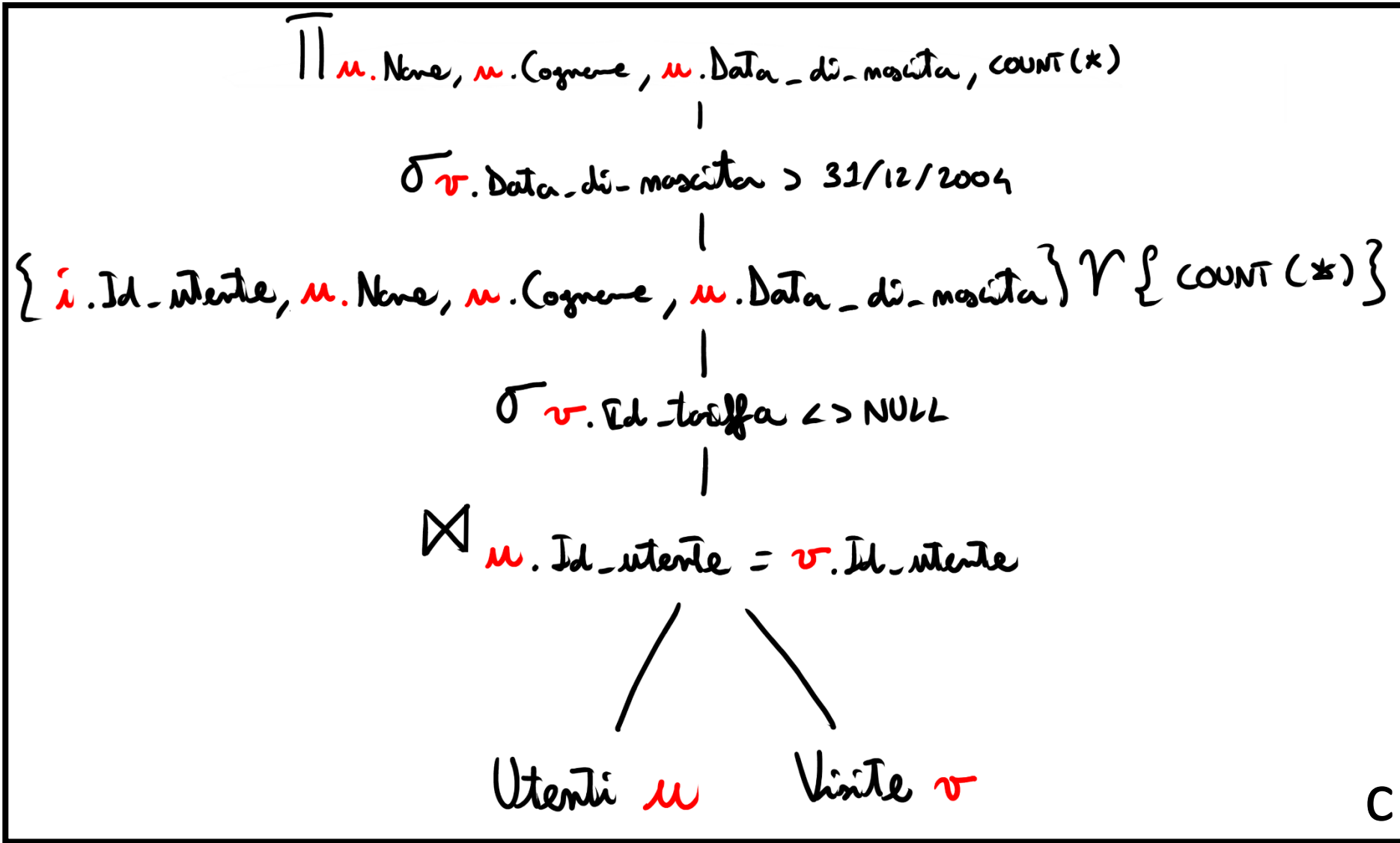
5. Piani di accesso

I. Scrivere un piano di accesso logico delle query a), b), c);

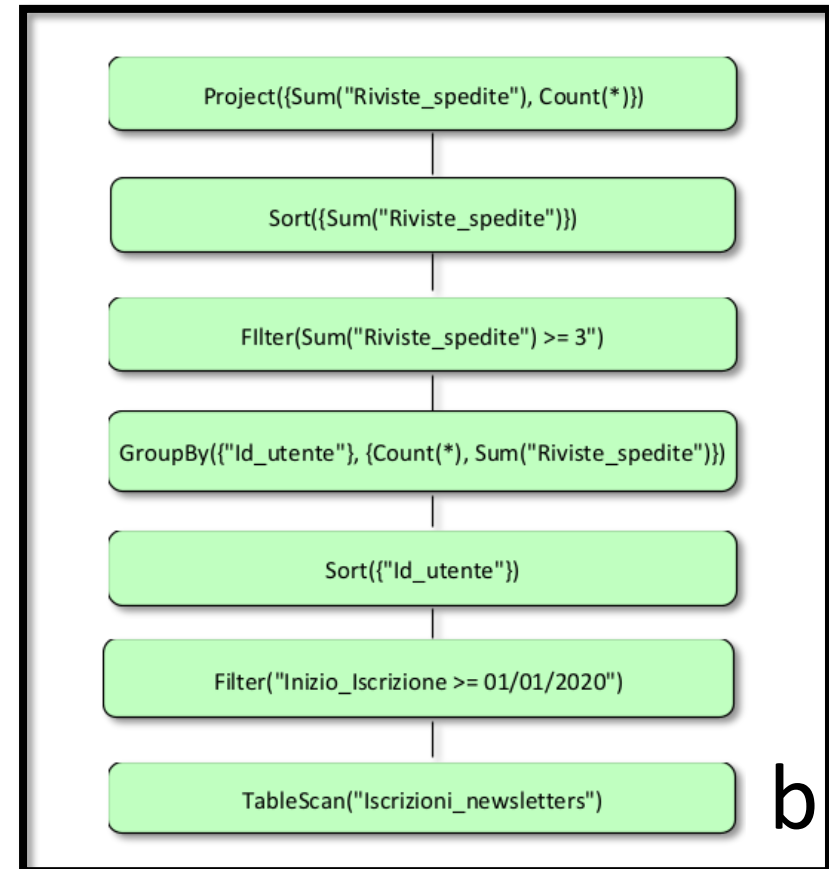
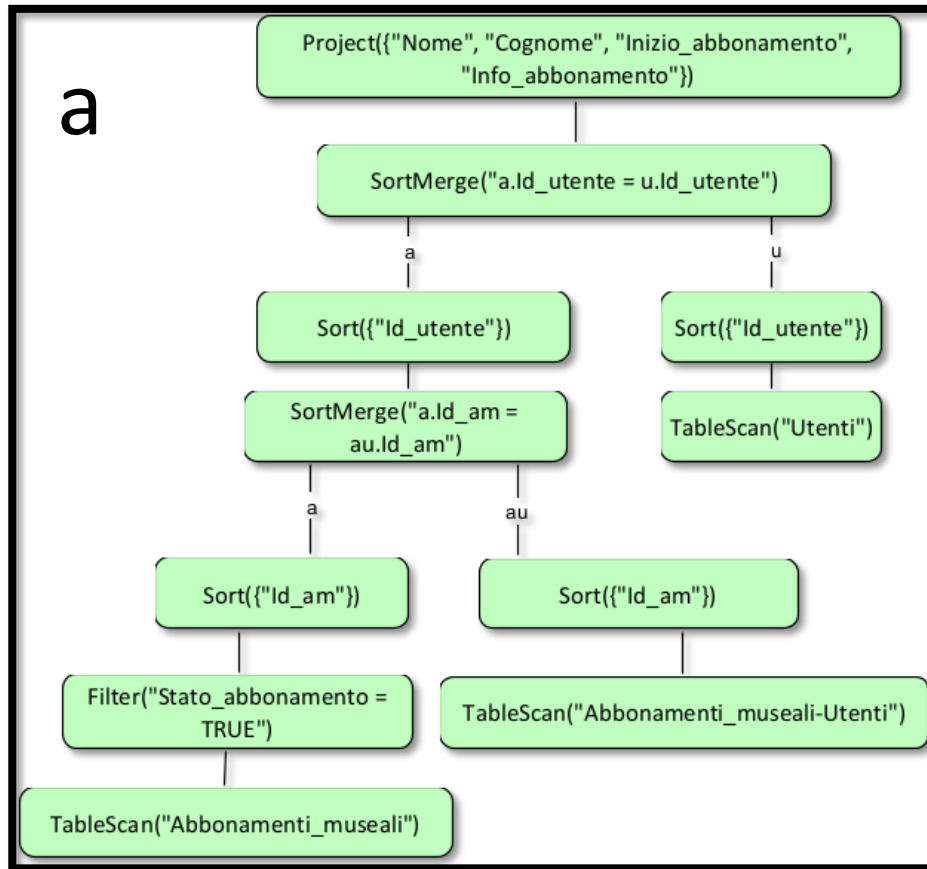


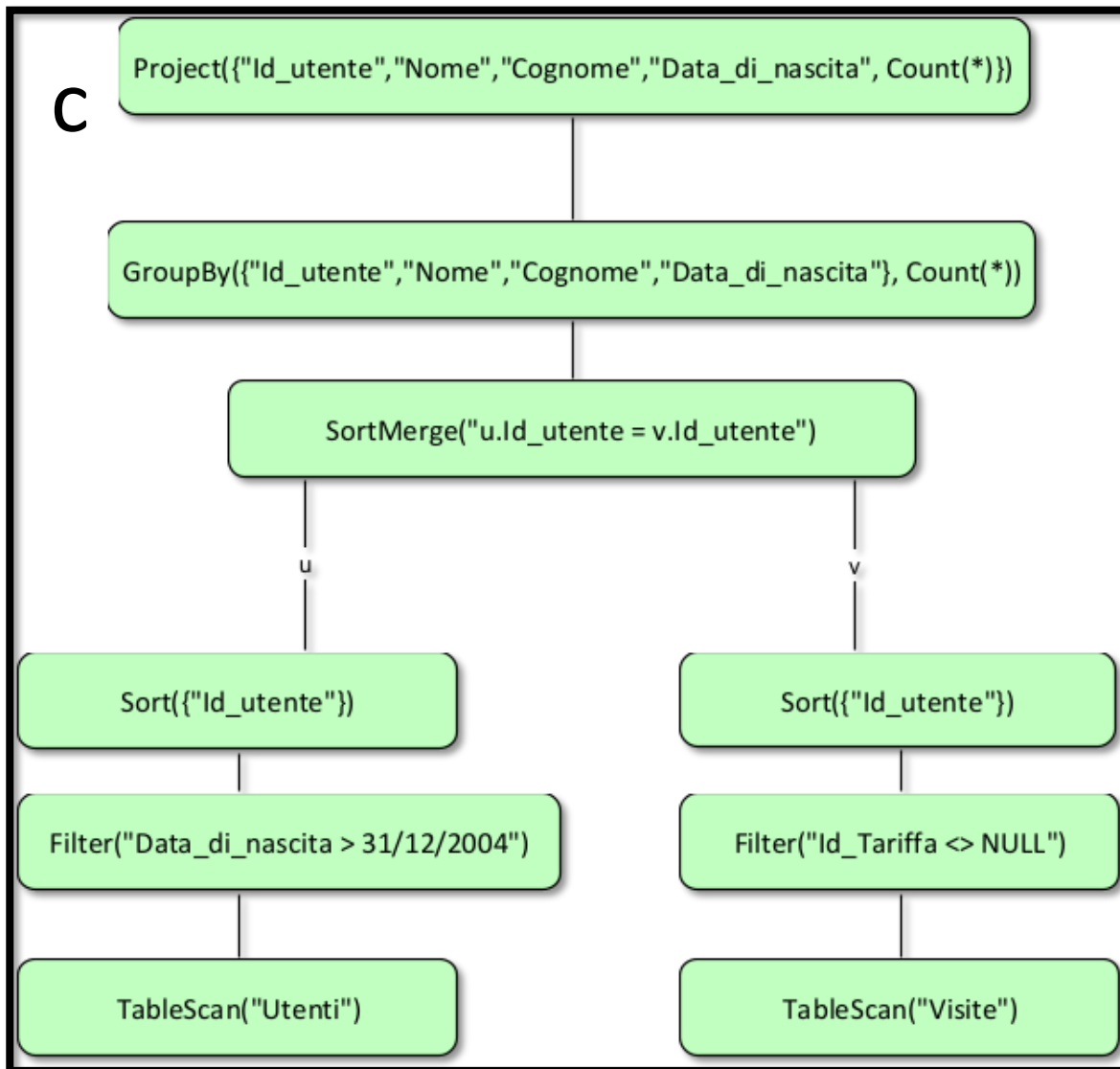


b



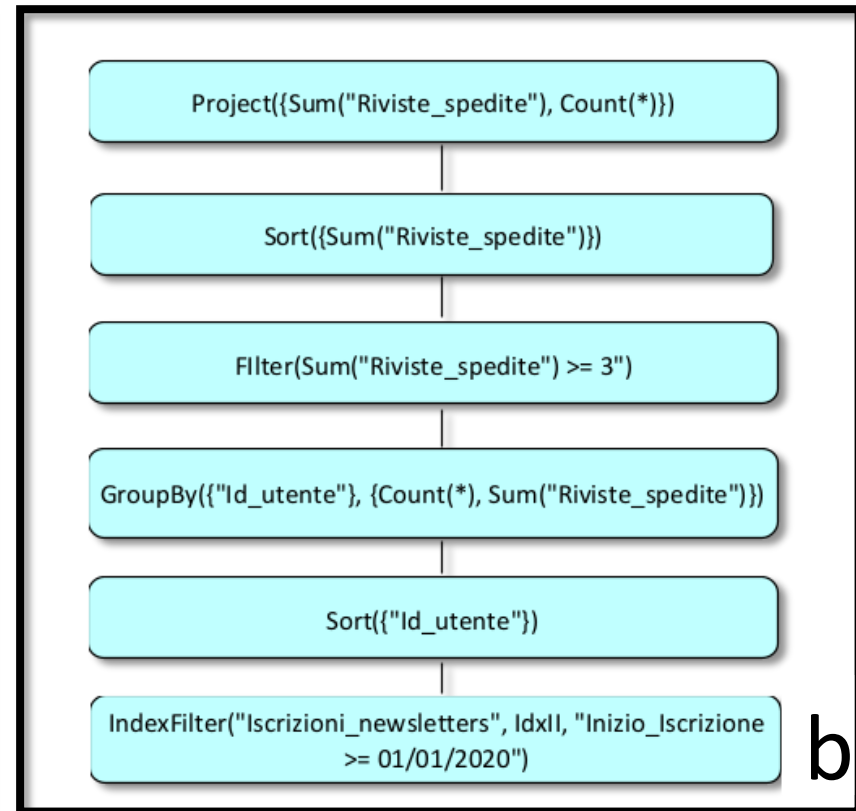
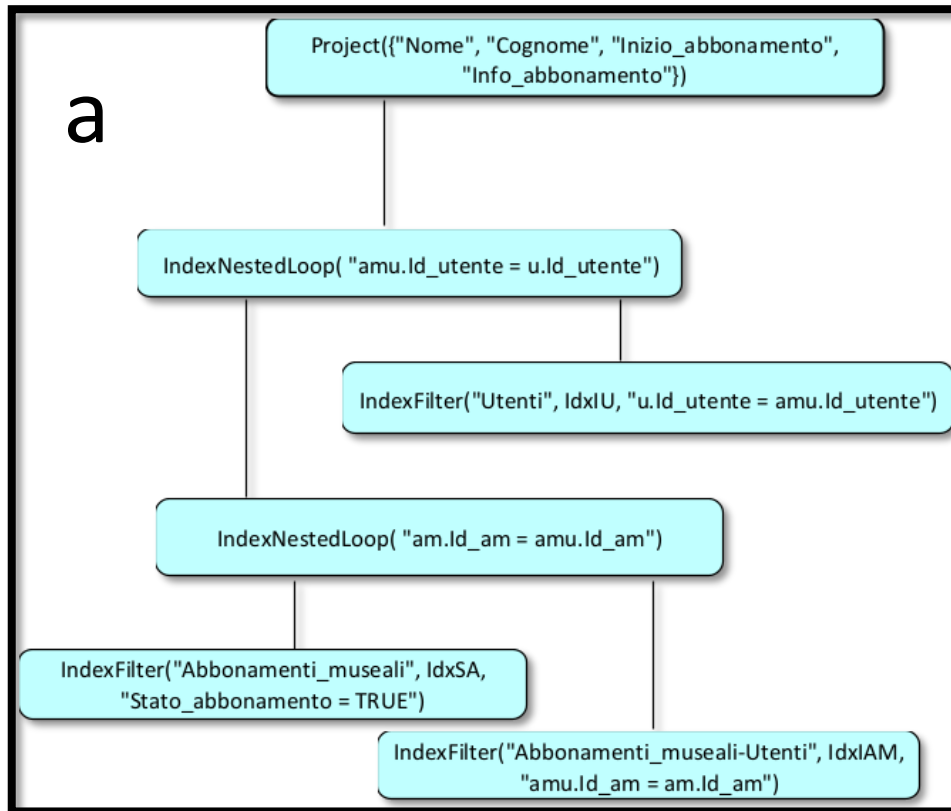
II. Scrivere un piano di accesso fisico efficiente per i tre piani di accesso logico al punto I che non fanno uso di indici, e (opzionale) verificare se la sort prima della Group By può essere evitata;

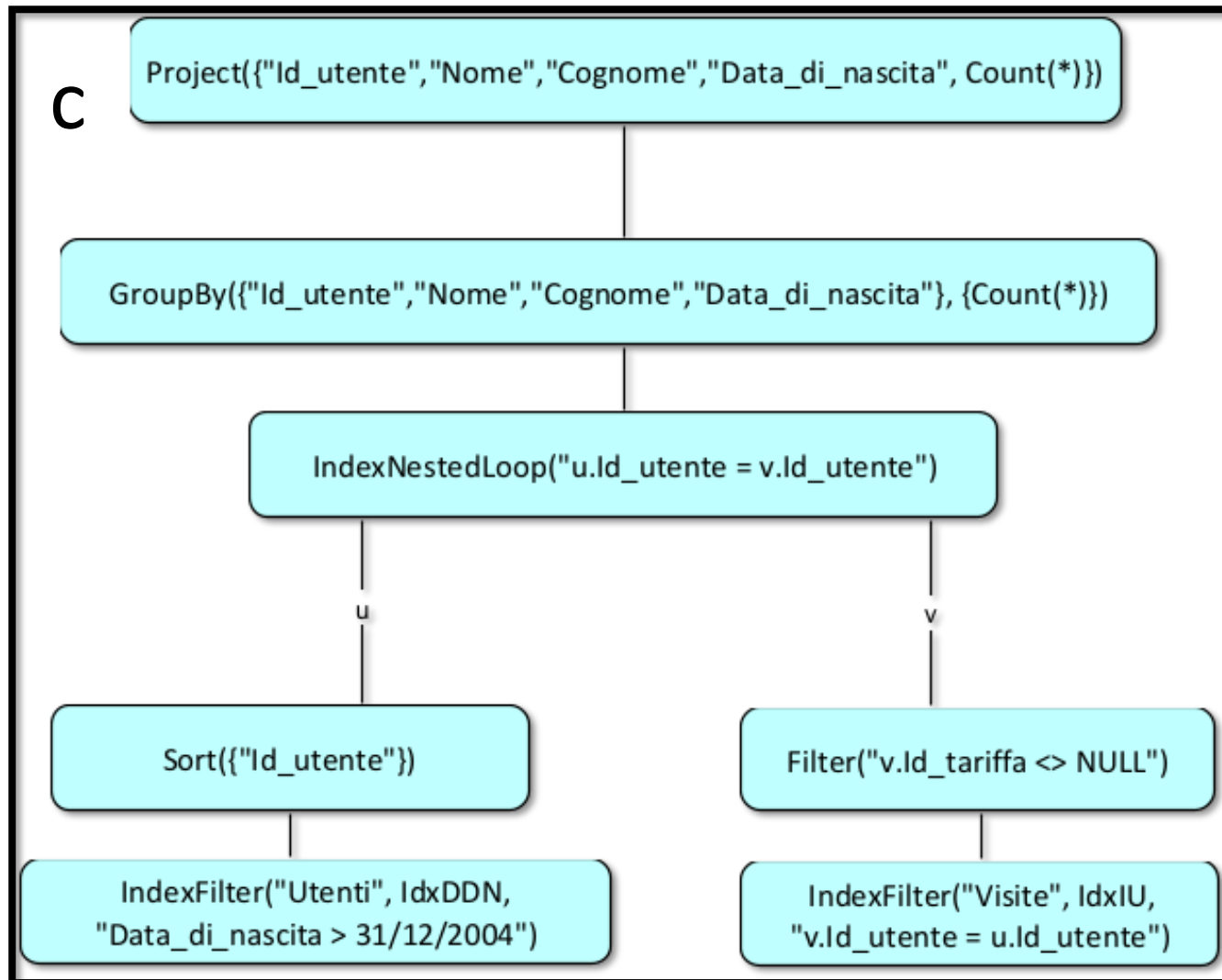




Nota : in entrambi i piani di accesso fisici b e c è presente una Sort che precede la Group By. Nel b è posta direttamente prima e nel c le due precedono la Sort Merge che mantiene l'ordinamento dei record degli operatori. Nel piano di accesso c è impossibile evitare l'ordinamento dei record, anche per necessità da parte della giunzione, a meno che in memoria siano memorizzati in modo sequenziale rispetto all'attributo "Id_utente". Anche nel piano di accesso b è impossibile evitare la Sort a meno dell'ordinamento sequenziale.

III. Scrivere un piano di accesso fisico efficiente per i tre piani di accesso logico al punto I che fanno uso di due indici (o comunque del numero massimo di indici possibili), e (opzionale) verificare se la sort prima della Group By può essere evitata.





Nota: In questo caso le riflessioni relative alle Sort che precedono le Group By fatte in precedenza non sono più valide: nel piano di accesso b la Index Filter non mantiene l'ordinamento dei record in memoria, quindi è necessaria una Sort; Nel piano di accesso c la Sort è necessaria per lo stesso motivo, ma è stato possibile spostarla prima della Index Nested Loop perché questa mantiene l'ordinamento dei record dell'operatore sinistro (operatore esterno).