

1. Descrizione del dominio

L'Utente rappresenta ogni persona iscritta alla piattaforma e può creare gruppi classe per rimettersi in contatto con i suoi vecchi compagni/professori. Ogni iscritto può partecipare a uno o più gruppi e può essere di due tipi:

- Ex-Alunno: può cercare gruppi di classe e chiedere di unirsi.
- Professore: può essere invitato in un gruppo classe e incluso nei dossier. Non può vedere commenti che lo riguardano.

Il GruppoClasse è composto da ex alunni e professori e ha specificati l'anno dell'esame di maturità, la sezione, il nome della scuola e la città. Può esistere un solo gruppo per ogni classe e creato da un solo utente che ne diventa automaticamente partecipante e amministratore. Quando il numero di partecipanti raggiunge i 6 chi ha creato il gruppo designa altri due amministratori. Se qualche amministratore esce dal gruppo i partecipanti ne designano un altro.

Per ciascuna persona da ritrovare di una classe viene creato un Dossier che raccoglie informazioni personali in forma testuale e materiali multimediali. È visibile, modificabile e cancellabile da tutti i partecipanti della classe e può contenere commenti scritti da altri utenti del gruppo.

Gli utenti del gruppo possono scrivere Commenti nei dossier.

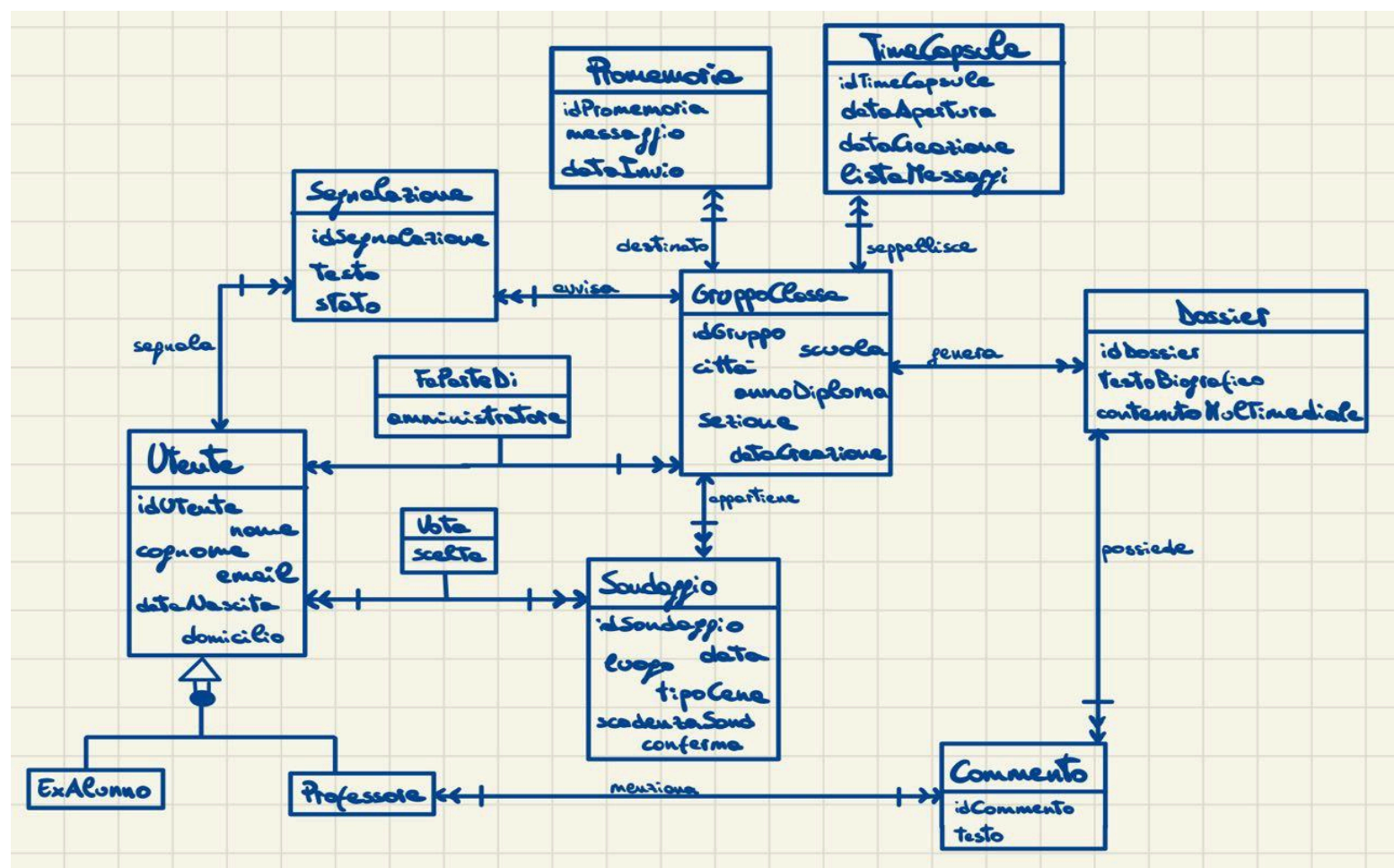
I Sondaggi vengono creati dagli amministratori del GruppoClasse per decidere dettagli della cena come data, luogo e tipo di cena.

I Promemoria sono messaggi automatici programmati dagli amministratori destinati al gruppo classe.

I TimeCapsule sono un archivio di messaggi associato ad un gruppo classe da rileggere dopo più di dieci anni. Ogni gruppo classe può avere più di una Time-Capsule.

Le Segnalazioni sono generate dal sistema e servono ad avvisare gruppi classe che ci sono utenti che potrebbero aver fatto parte della classe.

2. Schema Concettuale



Vincoli intrarelazionali

1. **GruppoClasse:**
 - a. Per ogni combinazione di scuola, città, anno del diploma e sezione, può esistere al più un solo GruppoClasse.
2. **Time Capsule:**
 - a. Una Time Capsule può essere aperta solo se sono trascorsi almeno dieci anni dalla data di creazione.

Vincoli interrelazionali

1. Amministratori del gruppo

- a. Ogni gruppo deve avere almeno un partecipante con ruolo di amministratore.
- b. Se il numero di partecipanti al gruppo è maggiore di sei, devono essere presenti esattamente tre amministratori.

2. Gestione dei sondaggi

- a. Solo i partecipanti con ruolo di amministratore possono creare sondaggi
- b. Solo gli amministratori possono confermare che una cena avrà luogo
- c. Dopo la conferma di un sondaggio, non è possibile crearne di nuovi prima che sia trascorso un anno dalla data di creazione del gruppo.
- d. Solo i membri del gruppo possono partecipare ai sondaggi esprimendo il proprio voto
- e. Ogni membro può votare al massimo una volta per ciascun sondaggio.

3. Gestione dei dossier

- a. Solo i membri del gruppo possono accedere, modificare o eliminare le informazioni contenute nei dossier.
- b. Solo i membri del gruppo possono aggiungere commenti ai dossier.

4. Professori

- a. I professori non possono leggere i commenti nei dossier in cui vengono esplicitamente menzionati.
- b. I professori non possono cercare gruppi classe né richiedere l'iscrizione: possono essere aggiunti solamente tramite invito da parte di un partecipante.

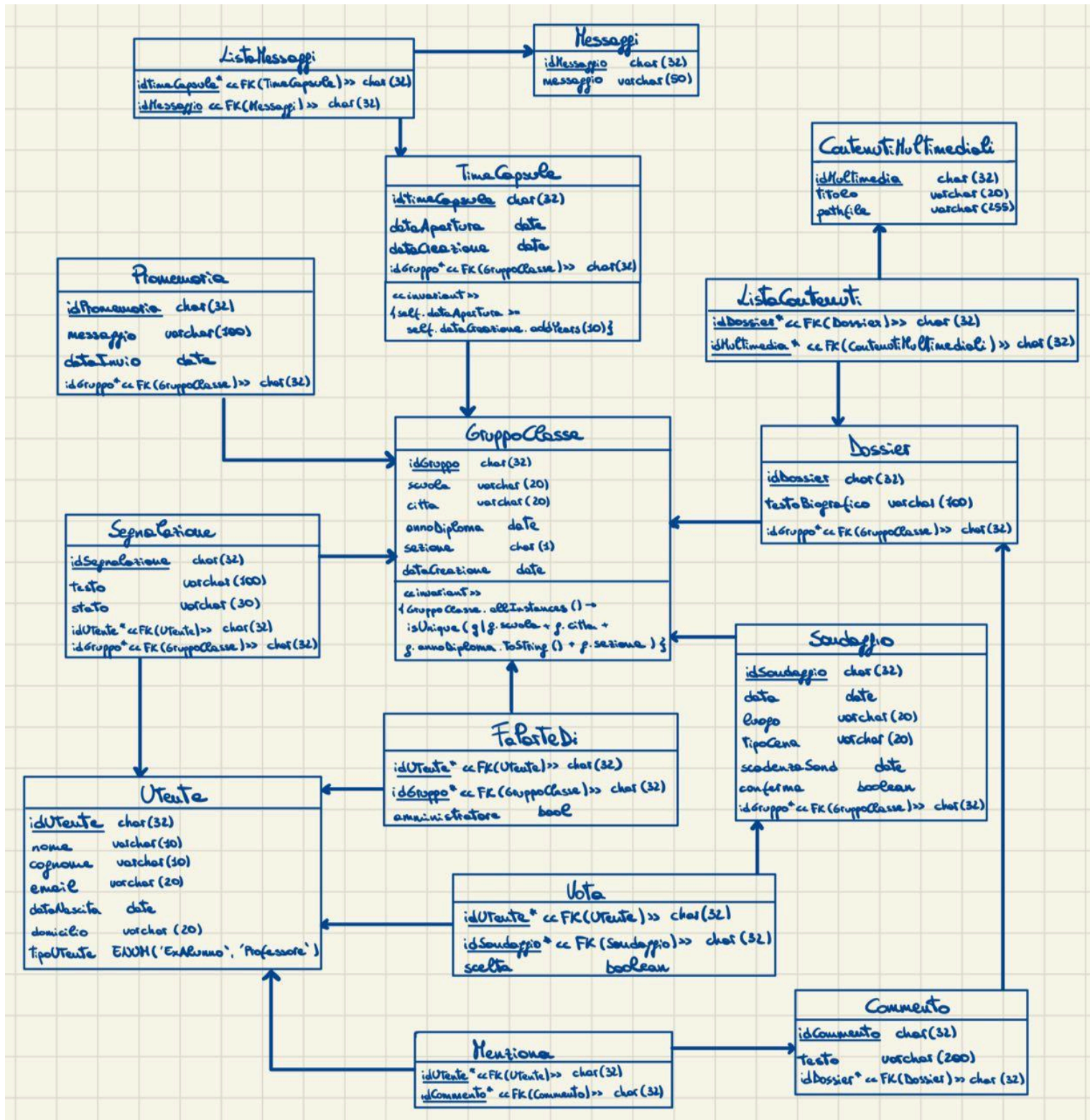
5. Promemoria

- a. Solo gli amministratori del gruppo possono scrivere promemoria.

6. Time Capsule

- a. Solo i membri del gruppo possono creare Time Capsule e aggiungervi messaggi.

3. Schema Logico-Relazionale



Testuale

Utente (

idUtente: char(32),
nome: varchar(10),
cognome: varchar(10),
email: varchar(50),
dataNascita: date,
domicilio: varchar(20),
tipoUtente: enum("ExAlunno", "Professore")
)

FaParteDi (

idUtente*: char(32),
idGruppo*: char(32),
amministratore: boolean
)

GruppoClasse (

idGruppo: char(32),
scuola: varchar(20),
citta: varchar(20),
annoDiploma: date,
sezione: char(1),
dataCreazione: date
)

Dossier (

idDossier: char(32),
testoBiografico: varchar(100),
idGruppo*: char(32)
)

Commento (

idCommento: char(32),
testo: varchar(200),

Segnalazione (

idSegnalazione : char(32),
testo: varchar(100),
stato: varchar(30),
idUtente*: char(32),
idGruppo*: char(32)
)

Promemoria (

idPromemoria : char(32),
messaggio: varchar(100),
dataInvio: date,
idGruppo* : char(32)
)

TimeCapsule (

idTimeCapsule : char(32),
dataApertura: date,
dataCreazione: date,
idGruppo* : char(32)
)

ListaMessaggi (

idTimeCapsule* : char(32),
idMessaggio* : char(32)
)

Messaggi (

idMessaggio : char(32),
messaggio: varchar(50)
)

idDossier*: char(32)
)

Menziona (
idUtente* : char(32),
idCommento* : char(32)
)

Sondaggio (
idSondaggio : char(32),
data: date,
luogo: varchar(20),
tipoCena: varchar(20),
scadenzaSond: date,
conferma: boolean,
idGruppo* : char(32)
)

Vota (
idUtente* : char(32),
idSondaggio* : char(32),
scelta: boolean
)

ContenutiMultimediali (
idMultimedia: char(32),
titolo: varchar(20),
pathFile: varchar(255)
)

ListaContenuti (
idDossier* : char(32),
idMultimedia* : char(32)
)

Dipendenze Funzionali

1. Utente

- idUtente → nome, cognome, email, dataNascita, domicilio, tipoUtente

2. FaParteDi

- (idUtente, idGruppo) → amministratore

3. GruppoClasse

- idGruppo → scuola, città, annoDiploma, sezione, dataCreazione

4. Dossier

- idDossier → testoBiografico, idGruppo

5. Commento

- idCommento → testo, idDossier

6. Sondaggio

- idSondaggio → data, luogo, tipoCena, scadenzaSond, conferma, idGruppo

7. Vota

- (idUtente, idSondaggio) → scelta

8. Segnalazione

- idSegnalazione → testo, stato, idUtente, idGruppo

9. Promemoria

- idPromemoria → messaggio, dataInvio, idGruppo

10. TimeCapsule

- idTimeCapsule → dataApertura, dataCreazione, idGruppo

11. Messaggi

- idMessaggio → messaggio

12. ContenutiMultimediali

- idMultimedia → titolo, pathFile

Tutte le dipendenze funzionali rispettano la forma normale di Boyce Codd

4. SQL

A)

uso di proiezione, join e restrizione;

Seleziona i diversi luoghi dei sondaggi dei gruppi classe che hanno come sezione la E

```
SELECT DISTINCT s.luogo
FROM Sondaggio s
JOIN GruppoClasse g ON s.idGruppo=g.idGruppo
WHERE g.sezione= 'E'
```

B)

uso di group by con having, where e sort. Si richiede inoltre che la relativa lista degli attributi dopo il costrutto SELECT deve contenere almeno 3 elementi, due dei quali devono essere attributi della tabella;

Seleziona tutti gli utenti di nome Lorenzo che condividono lo stesso cognome con almeno un altro omonimo, mostrando il numero di occorrenze per ciascuna combinazione nome-cognome. I risultati sono ordinati in ordine alfabetico crescente per cognome.

```
SELECT nome, cognome, COUNT(*) AS numeroOmonimi
FROM Utente
WHERE nome = 'Lorenzo'
GROUP BY nome, cognome
HAVING COUNT(*) >= 2
ORDER BY cognome ASC
```

C)

uso di join, group by con having e where;

Restituisce gli ID degli utenti che hanno almeno 3 segnalazioni con stato "in elaborazione", insieme al numero di tali segnalazioni.


```
SELECT idUtente, COUNT (*) AS numeroSegnalazioni
FROM Utente u
JOIN Segnalazione s ON s.idUtente=u.idUtente
WHERE s.stato = 'in elaborazione'
GROUP BY u.idUtente
HAVING COUNT(*) >= 3
```

D)

uso di select annidata con quantificazione esistenziale

Seleziona gli ID dei dossier che hanno almeno un commento associato.

```
SELECT idDossier
FROM Dossier d
WHERE EXIST (
    SELECT 1
    FROM Commento c
    WHERE c.idDossier=d.idDossier
)
```

E)

uso di select annidata con quantificazione universale;

Seleziona gli ID dei gruppi classe per cui tutti i sondaggi associati sono stati confermati

```
SELECT g.idGruppoClasse
FROM GruppoClasse g
WHERE NOT EXISTS (
    SELECT *
    FROM Sondaggio s
    WHERE s.idGruppo = g.idGruppo
    AND s.conferma <> TRUE
)
```

F)

uso di subquery di confronto quantificato usando una subquery.

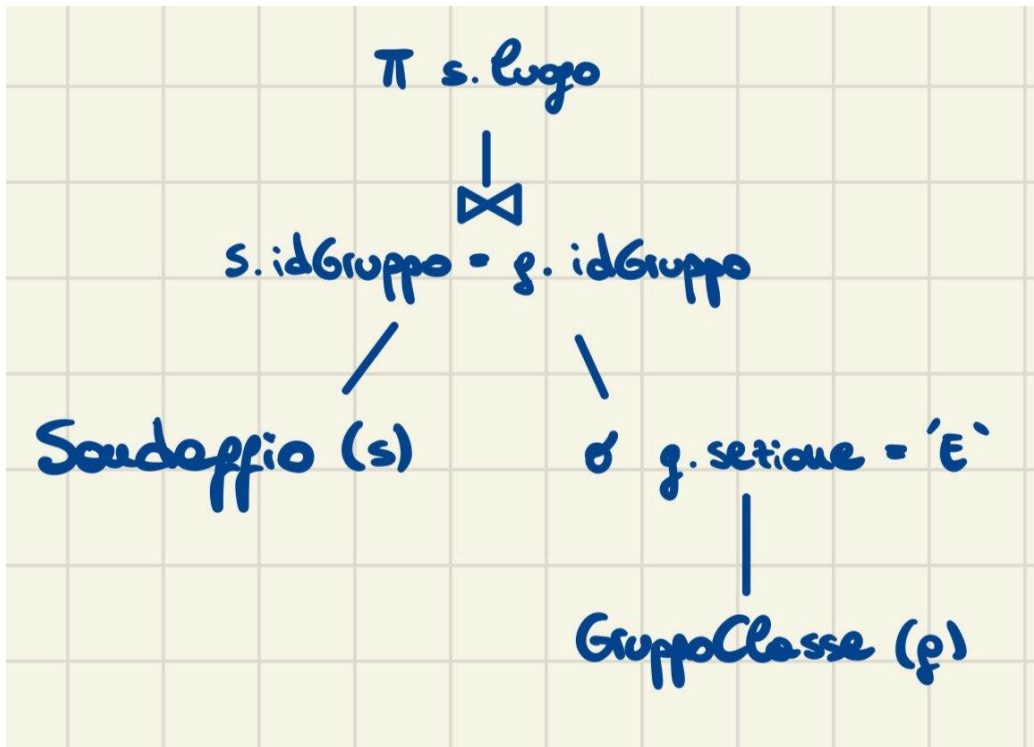
Restituisce i gruppi classe che hanno almeno un sondaggio con tipoCena pizza

```
SELECT *  
FROM GruppoClasse g  
WHERE 'pizza' = ANY (  
    SELECT s.tipoCena  
    FROM Sondaggio s  
    WHERE g.idGruppo = s.idGruppo  
)
```

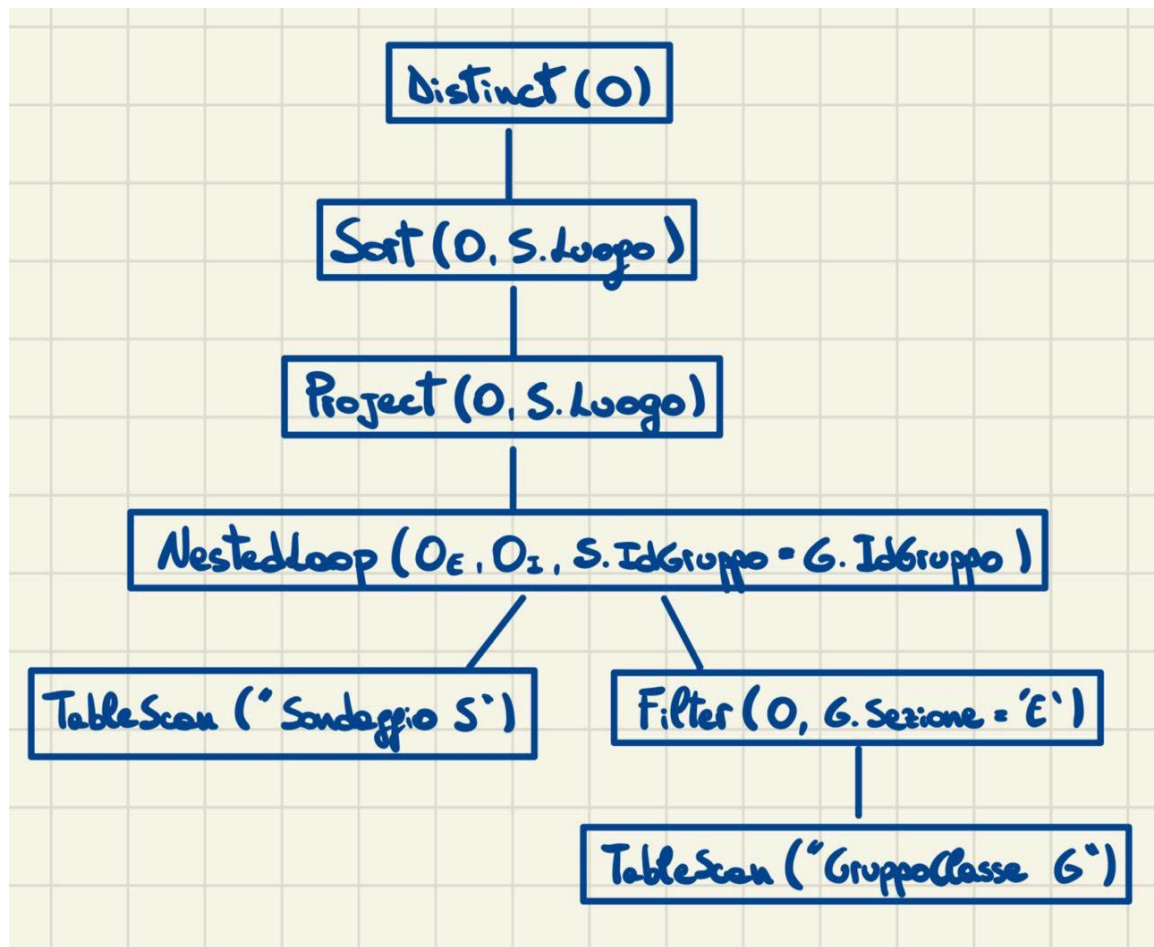
5. Piani di Accesso

A)

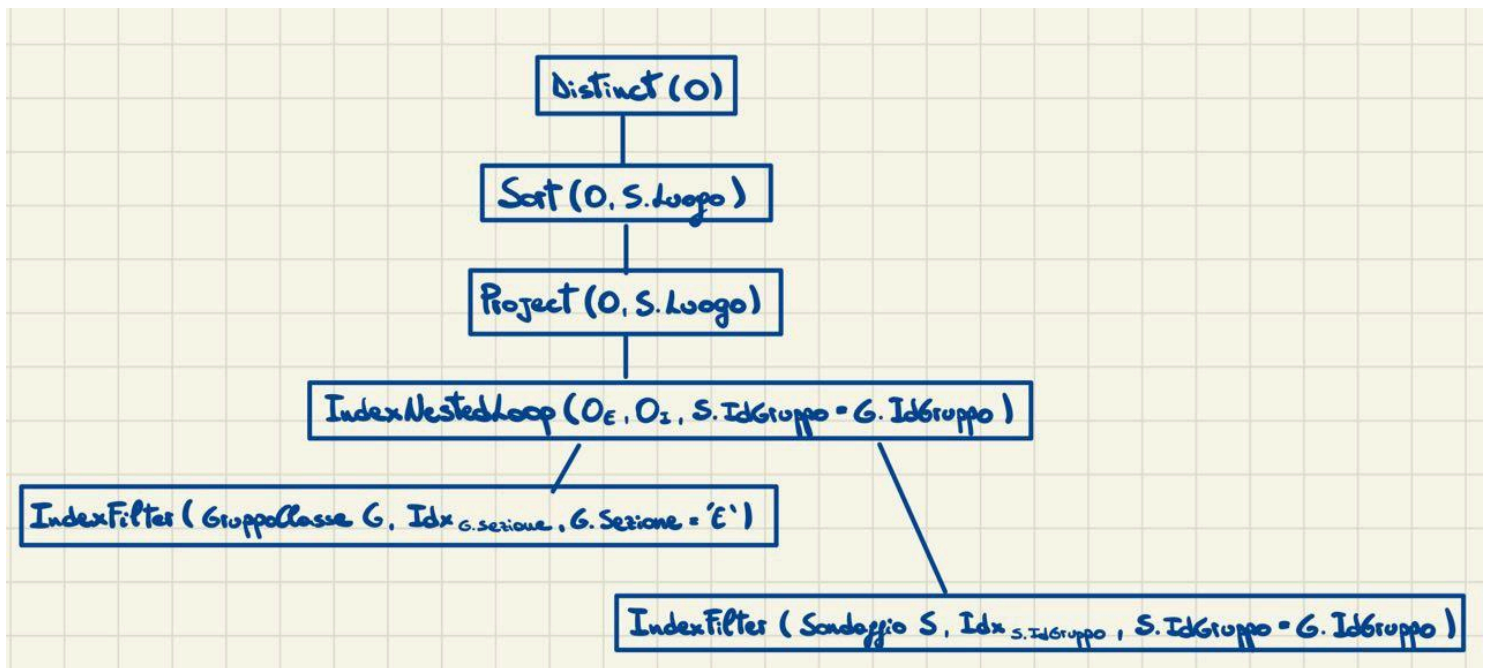
logico



fisico senza indici

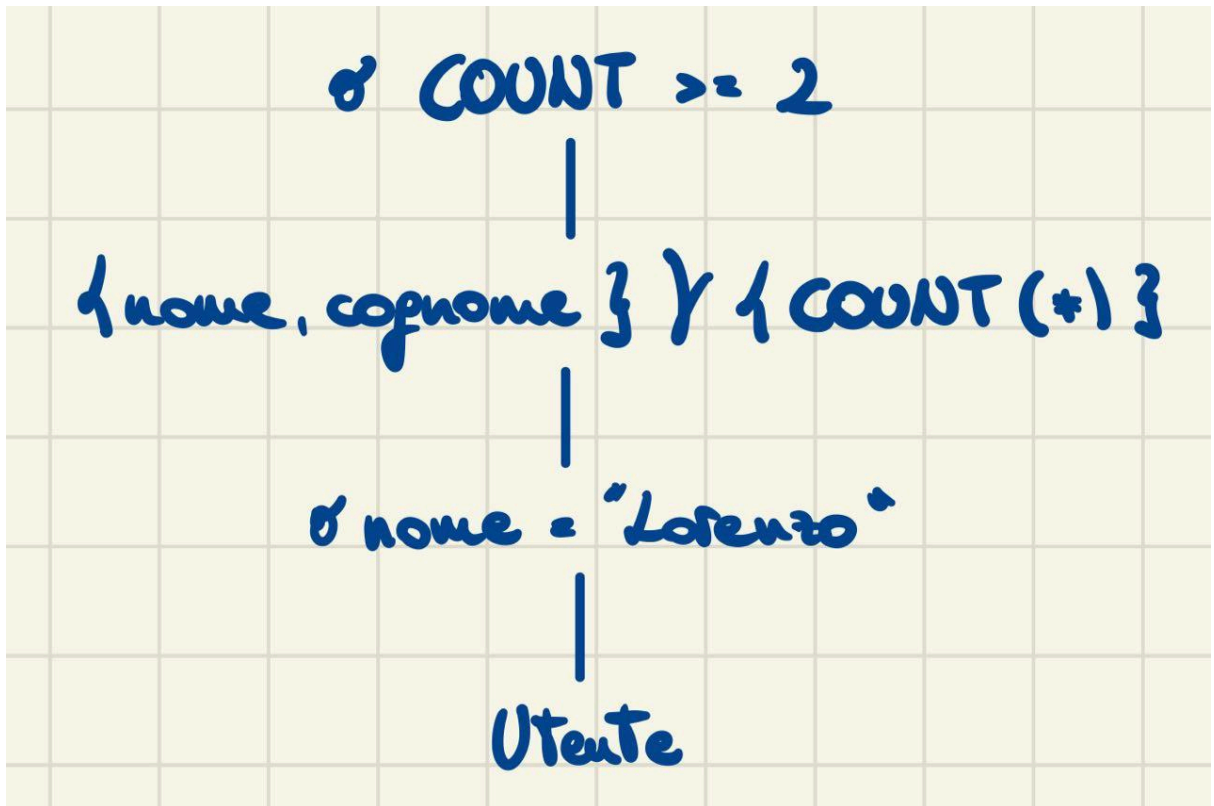


fisico con indici

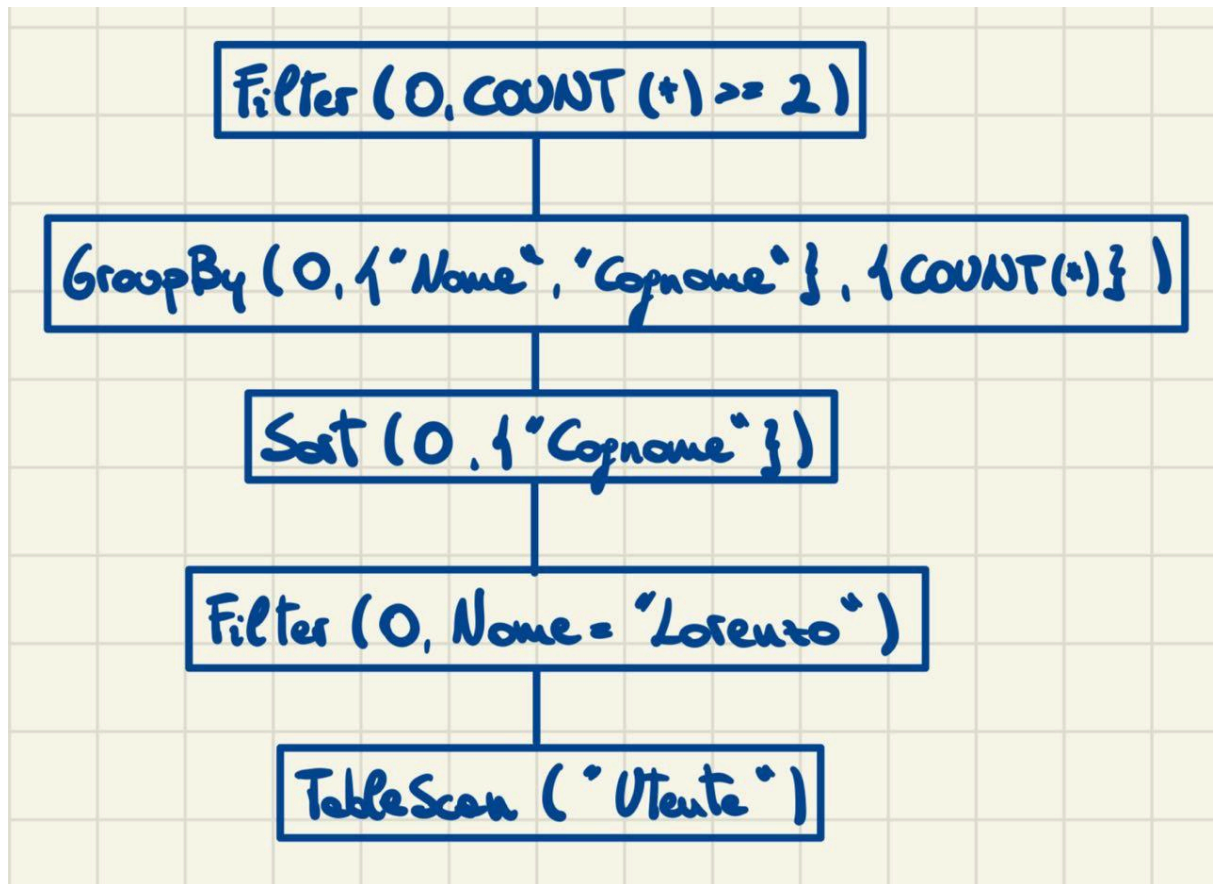


B)

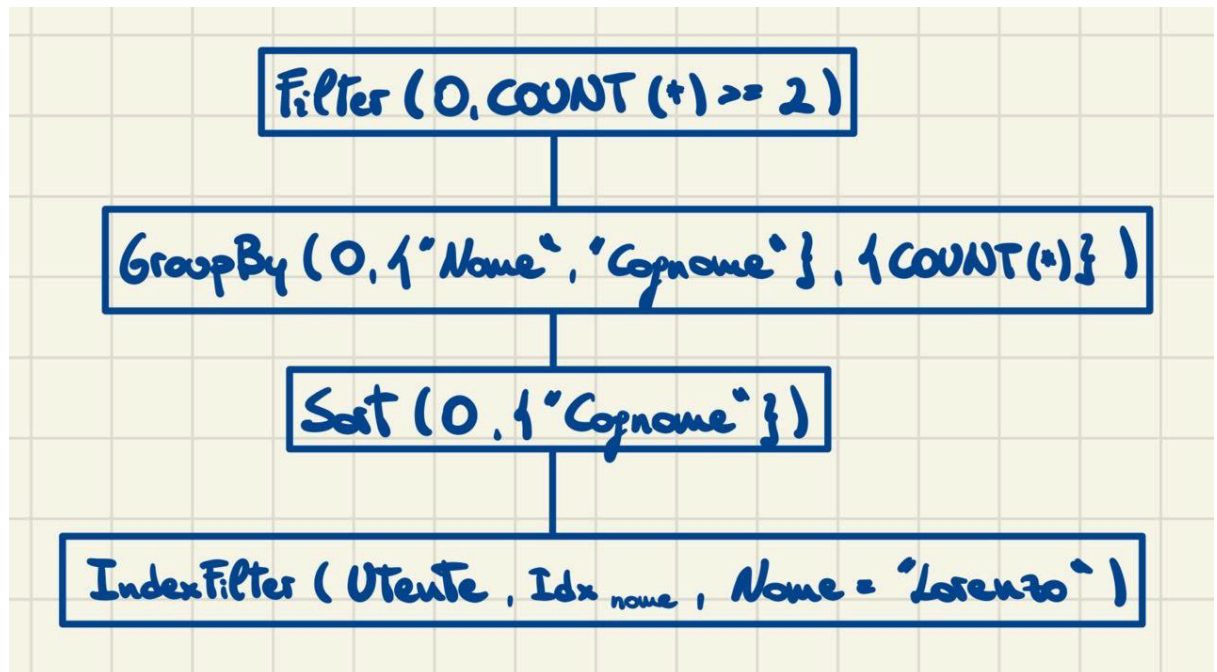
Logico



Fisico senza indici

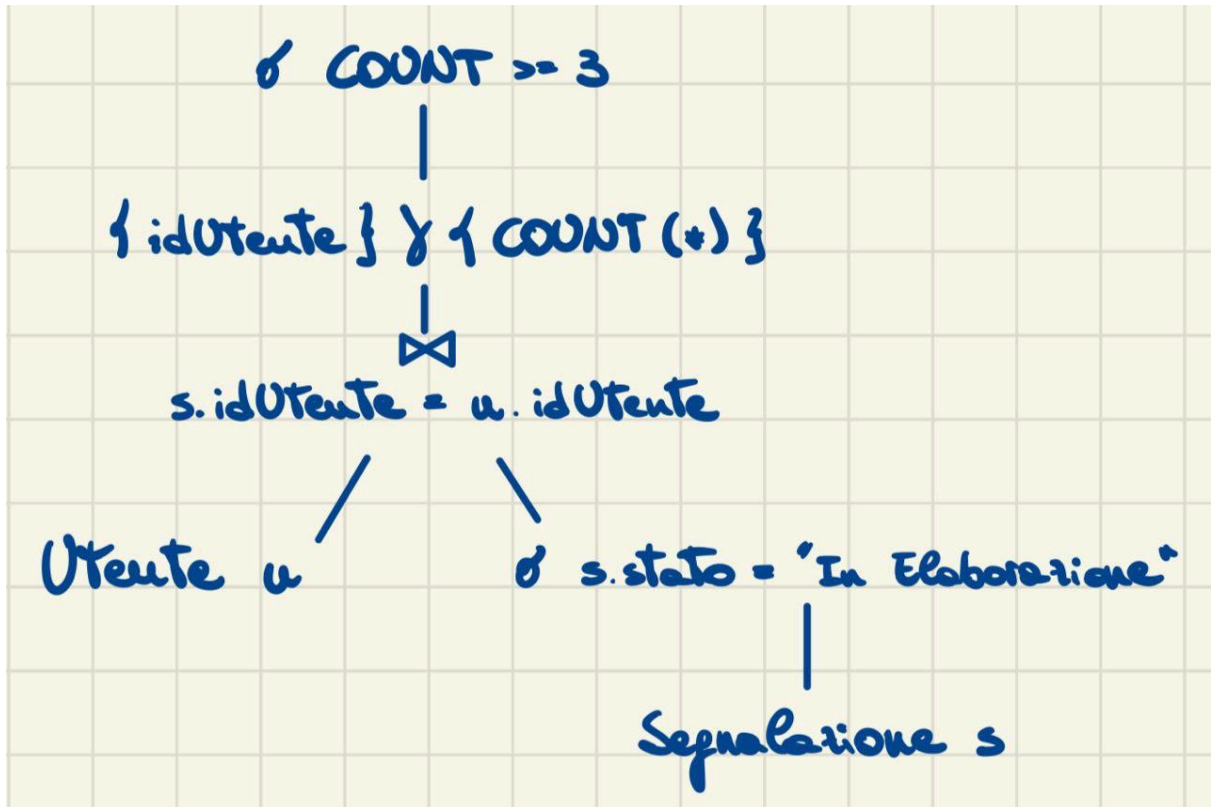


Fisicio con indici

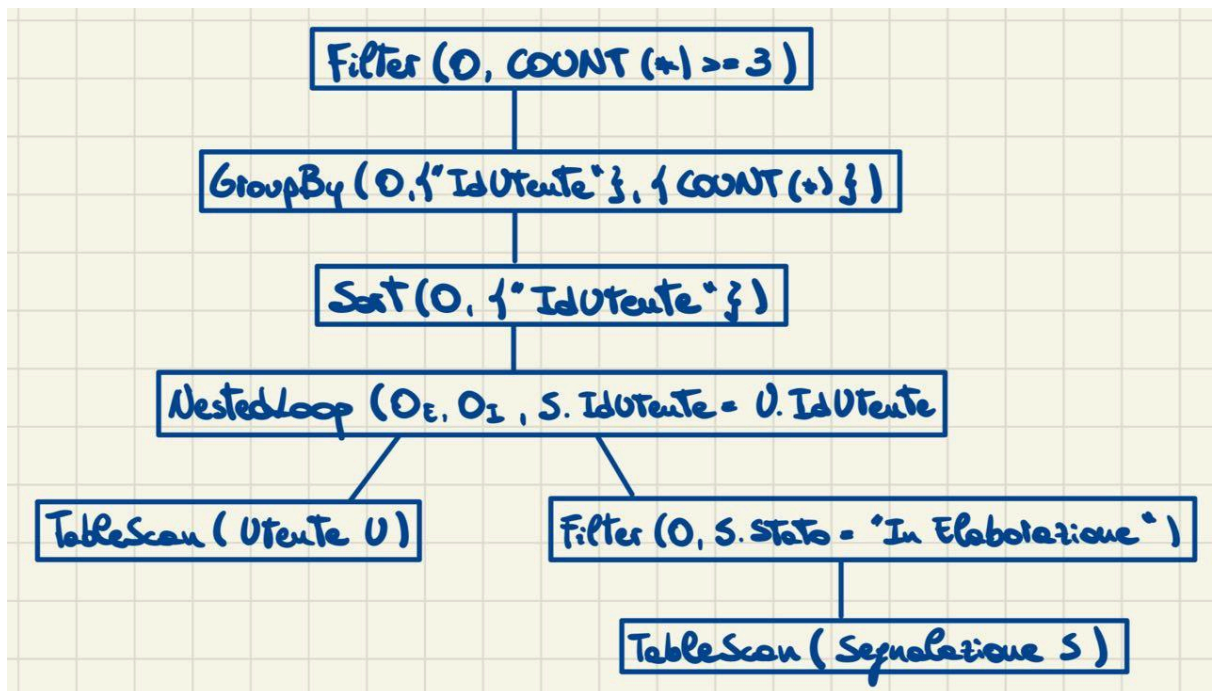


C)

Logico



fisico senza indici



fisico con indici

