

Progetto di Basi di Dati

Corso di laurea in Informatica

Dipartimento di Informatica

Appello 1

data di consegna: 30 maggio 2022



UNIVERSITÀ
DI PISA

Il fascino discreto dei musei dell'Università di Pisa



Antonio Cirielli - 600375

Alessandro Querci - 578615

Marcello Satta - 580495

1. Descrizione del dominio

Nell'analisi delle specifiche di questo progetto siamo partiti dall'identificare tre classi principali: la classe **Utenti**, la classe **Musei**, la classe **Biglietti**. Da queste tre classi abbiamo derivato, da una parte i tipi di biglietti messi a disposizione dal sistema (abbonamento o biglietto semplice), dall'altra i tipi di musei presenti (gratuiti e a pagamento). Segue la descrizione dettagliata di tutte le classi individuate.

Utente: persone fisiche che acquistano uno o più biglietti o sottoscrivono un abbonamento, dichiarando il proprio nome e cognome.

Opzionalmente è possibile dichiarare un recapito composto da numero di telefono, e-mail e comune di residenza.

Eventualmente, gli utenti possono ottenere uno o più buoni spendibili presso l'UnipiStore.

Biglietto: biglietto fisico attivo (non scaduto) che viene acquistato dagli utenti, di cui ci interessa l'intestatario, la data di emissione, il prezzo e se è un biglietto Abbonamento, Combo o Singolo

Abbonamento: può essere settimanale (Combo3 nelle specifiche) e sottoscrivibile da tutti gli utenti oppure annuale, se e solo se l'utente ha dichiarato la propria residenza presso il Comune di Pisa.

Biglietto Combo: tipo di biglietto che combina gli accessi a due musei. Abbiamo due tipi di biglietti combo, uno che permette di visitare due musei grandi e uno che permette di visitare un museo grande e uno piccolo.

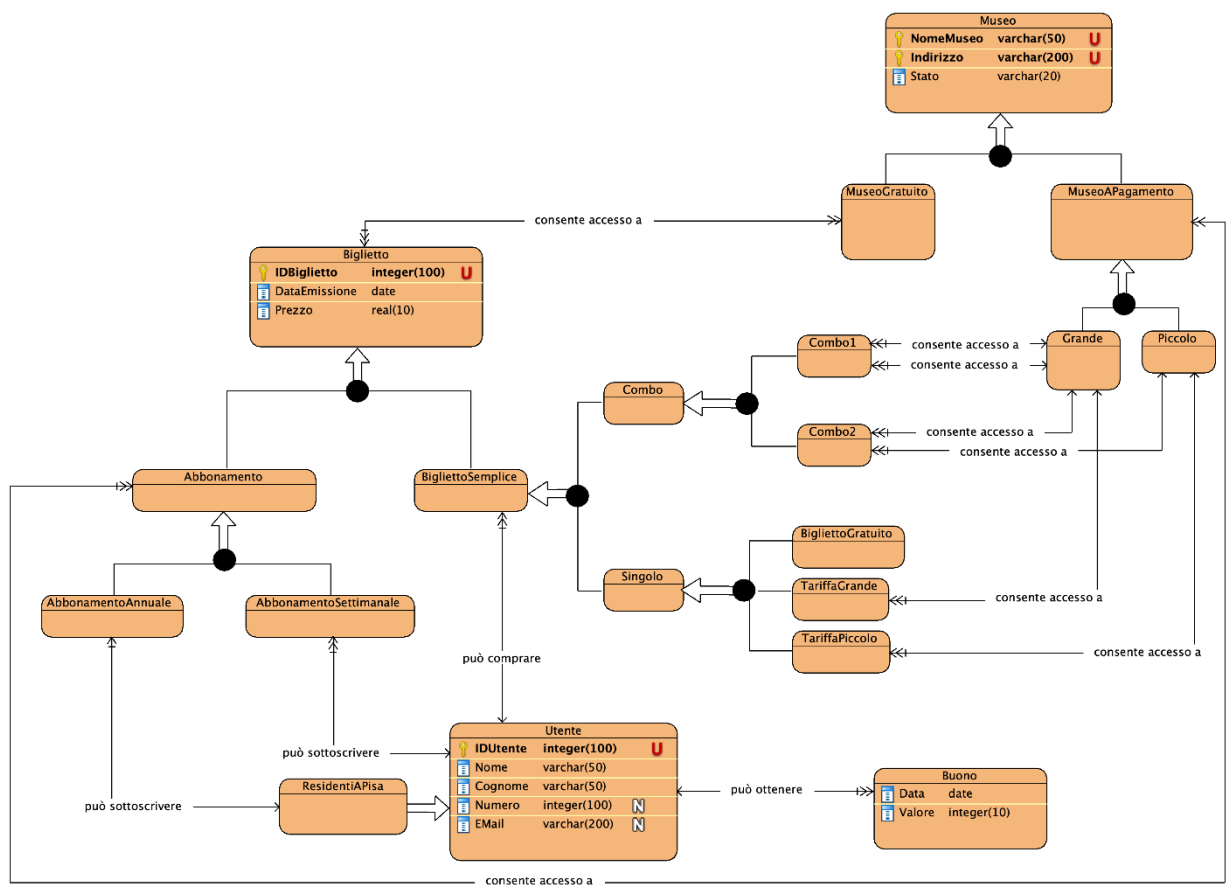
Biglietto Singolo: biglietto che permette l'accesso ad un solo museo tra quelli presenti, si differenziano tra loro per il costo e a costi diversi corrispondono tipi di musei diversi a cui il biglietto stesso consente l'accesso.

Museo: il museo è il luogo fisico in cui gli utenti si recano per prendere parte alle esposizioni presenti. Si distinguono tra loro per nome e indirizzo, grandezza e costo del biglietto. Alcuni musei possono essere chiusi. Le mostre temporanee vengono considerate come Museo di tipo "grande" e quando non sono disponibili vengono considerate come museo "chiuso".

Possibili operazioni e loro frequenza:

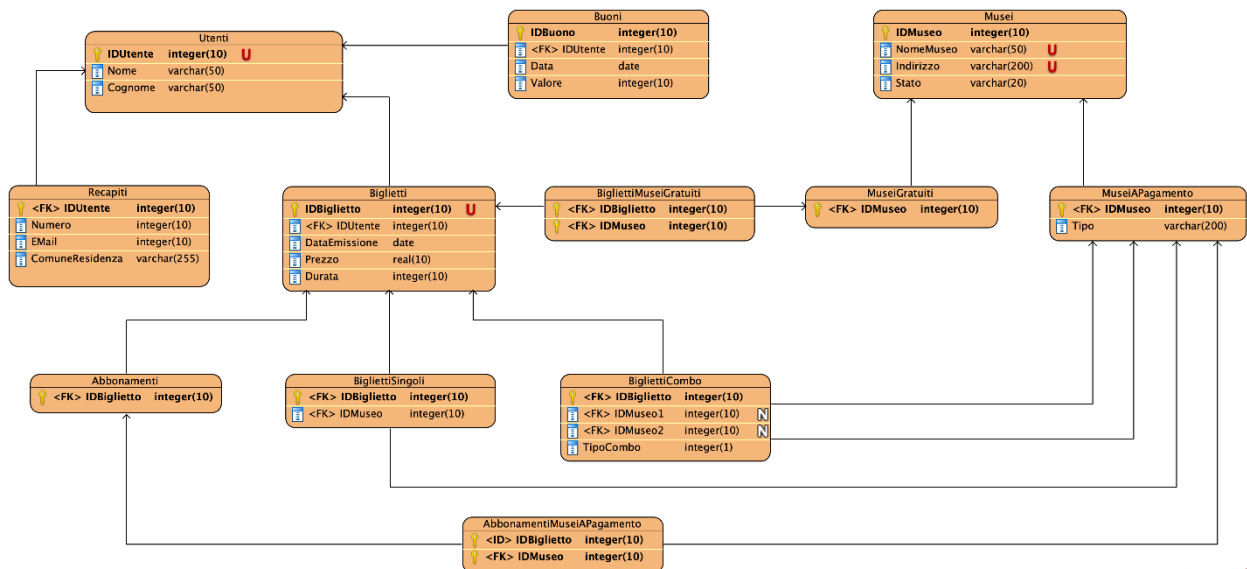
- Acquisto di un nuovo biglietto singolo da parte di un utente (200 operazioni al giorno);
- aggiunta di un nuovo museo ad accesso gratuito (una volta ogni anno);
- sottoscrizione di un abbonamento da parte di un utente residente nel Comune di Pisa (5 volte al giorno);
- stampa di tutti i biglietti attivi in un determinato momento (1 volta al giorno);
- visualizzare l'anagrafica di tutti gli utenti che posseggono un abbonamento annuale attivo;
- visualizzare i musei grandi a cui l'accesso è garantito tramite i biglietti di tipo combo1 attivi, ordinati per numero di utenti che hanno acquistato il biglietto (1 volta al mese);

2. Schema concettuale



3.1. Schema logico relazionale in formato grafico

Le chiavi sono specificate tramite l'icona della chiave color giallo, le chiavi esterne invece, con il prefisso <FK>. La “N” specifica il fatto che un attributo è Nullable, ovvero può assumere il valore NULL.



3.2 Schema logico relazionale in formato testuale

Le chiavi primarie sono sottolineate, le chiavi esterne sono identificate con un asterisco subito dopo il nome dell'attributo. I tipi degli attributi sono specificati sullo schema logico in formato grafico. Sotto ogni relazione abbiamo indicato, oltre alle dipendenze funzionali, alcuni vincoli intrarelazionali legati alle singole tabelle. Infine, sono presenti i vincoli interrelazionali identificati.

Non ci è sembrato necessario ripetere i vincoli intrarelazionali degli attributi presenti in altre relazioni come chiave esterna.

Tutte le relazioni rispettano la forma normale di Boyce-Codd.

- **Utenti** (IDUtente, Nome, Cognome)

DF: IDUtente -> Nome, Cognome

Vincoli intrarelazionali:

$$\forall u \in Utenti \mid u.IDUtente \in \mathbb{N} \setminus \{0\}$$

- **Recapiti** (IDUtente*, Numero, Email, ComuneResidenza)

DF: IDUtente -> IDUtente, Numero, Email, ComuneResidenza

Vincoli intrarelazionali:

$$\forall r \in Recapiti \mid r.Numero \in \mathbb{N} \setminus \{0\}$$

- **Buoni** (IDBuono, IDUtente*, Data, Valore)

DF: IDBuono -> IDUtente, Data, Valore

Vincoli intrarelazionali:

$$\forall b \in Buoni \mid b.IDBuono \in \mathbb{N} \setminus \{0\} \quad \wedge \quad \forall b \in Buoni \mid b.Valore = 3$$

- **Musei** (IDMuseo, NomeMuseo, Indirizzo, Stato)

DF: IDMuseo -> NomeMuseo, Indirizzo, Stato

Vincoli intrarelazionali:

$$\begin{aligned} & \forall m \in Musei \mid m.IDMuseo \in \mathbb{N} \setminus \{0\} \quad \wedge \\ & \wedge \quad \forall m \in Musei \mid m.Stato \in \{ "Aperto", "Chiuso" \} \end{aligned}$$

- **MuseiGratuiti** (IDMuseo*)

DF: IDMuseo -> IDMuseo (banale)

- **MuseiAPagamento** (IDMuseo*, Tipo)

DF: IDMuseo -> Tipo

Vincoli intrarelazionali:

$$\forall m \in MuseiAPagamento \mid m.Tipo \in \{ "Piccolo", "Grande" \}$$

- **Biglietti** (IDBiglietto, IDUtente*, DataEmissione, Prezzo, Durata)

IDBiglietto -> IDUtente, DataEmissione, Prezzo, Durata

Vincoli intrarelazionali:

$$\begin{aligned} & \forall b \in \text{Biglietti} \mid b.\text{IDBiglietto} \in \mathbb{N} \setminus \{0\} \quad \wedge \\ & \wedge \quad \forall b \in \text{Biglietti} \mid b.\text{Prezzo} \in \{2.5, 4, 5, 7, 10, 30\} \quad \wedge \\ & \wedge \quad \forall b \in \text{Biglietti} \mid b.\text{Durata} \in \{7, 365\} \end{aligned}$$

- **BigliettiMuseiGratuiti** (IDBiglietto*, IDMuseo*)

DF: IDBiglietto, IDMuseo -> IDBiglietto, IDMuseo (banale)

- **Abbonamenti** (IDBiglietto*)

DF: IDBiglietto -> IDBiglietto (banale)

- **AbbonamentiMuseiAPagamento** (IDBiglietto*, IDMuseo*)

DF: IDBiglietto, IDMuseo -> IDBiglietto, IDMuseo (banale)

- **BigliettiSingoli** (IDBiglietto*, IDMuseo*)

DF: IDBiglietto -> IDMuseo

- **BigliettiCombo** (IDBiglietto*, IDMuseo1*, IDMuseo2*, TipoCombo)

DF: IDBiglietto -> IDMuseo1, IDMuseo2, TipoCombo

Vincoli intrarelazionali:

$$\forall b \in \text{BigliettiCombo} \mid b.\text{TipoCombo} \in \{1, 2\}$$

Vincoli interrelazionali

- Per ogni nuovo abbonamento annuale con durata 365 giorni, l'utente che sottoscrive l'abbonamento è residente nel Comune di Pisa.

$$\forall a \in \text{Abbonamenti per cui } \exists b \in \text{Biglietti tale che } a.IDBiglietto = b.IDBiglietto \wedge$$
$$\wedge b.Durata = 365 \Rightarrow \exists u \in \text{Utenti}, r \in \text{Recapiti} \mid b.IDUtente = u.IDUtente \wedge$$
$$\wedge u.IDUtente = r.IDUtente \wedge r.ComuneDiResidenza = "PISA"$$

- Per ogni biglietto singolo con tariffa 2,5 €, questo permette l'accesso ad un solo museo, scelto tra l'insieme dei musei di tipo piccolo. Per ogni museo piccolo, il biglietto singolo che ne garantisce l'accesso costa 2,5 €.

Definizione insieme ausiliario:

$$\forall mp \in \text{MuseiAPagamento } \exists m \in \text{Musei tale che}$$
$$\text{tale che } \text{MuseiPiccoli} = \{mp.IDMuseo \text{ tale che } mp.Tipo = "Piccolo" \wedge mp.IDMuseo = m.IDMuseo\}$$

Vincolo:

$$\forall bs \in \text{BigliettiSingoli per cui } \exists b \in \text{Biglietti tale che}$$
$$\text{tale che } bs.IDBiglietto = b.IDBiglietto \wedge b.Prezzo = 2.5 \Leftrightarrow$$
$$\Leftrightarrow bs.IDMuseo \in \text{MuseiPiccoli}$$

- Per ogni biglietto singolo con tariffa 4 OR 5 €, questo permette l'accesso ad un solo museo, scelto tra l'insieme dei musei di tipo grande. Per ogni museo grande, il biglietto singolo che ne garantisce l'accesso costa 4 OR 5 €.

Definizione insieme ausiliario:

$$\forall mp \in \text{MuseiAPagamento } \exists m \in \text{Musei tale che}$$
$$\text{tale che } \text{MuseiGrandi} = \{mp.IDMuseo \text{ tale che } mp.Tipo = "Grande" \wedge mp.IDMuseo = m.IDMuseo\}$$

Vincolo:

$$\forall bs \in \text{BigliettiSingoli per cui } \exists b \in \text{Biglietti tale che}$$
$$\text{tale che } bs.IDBiglietto = b.IDBiglietto \wedge (b.Prezzo = 4 \vee$$
$$\vee b.Prezzo = 5) \Leftrightarrow bs.IDMuseo \in \text{MuseiGrandi}$$

- Per ogni biglietto combo di tipo 1, i musei a cui è garantito l'accesso sono 2 e appartengono entrambi all'insieme dei musei di tipo grande. Gli attributi possono essere NULL se l'utente non ha ancora scelto in quale museo entrare.

$\forall bc \in \text{BigliettiCombo}$ per cui $\exists b \in \text{Biglietti}$ tale che
 tale che $bc.IDBiglietto = b.IDBiglietto \wedge bc.TipoCombo = 1 \Leftrightarrow bc.IDMuseo1, bc.IDMuseo2 \in \text{MuseiGrandi}$

(Abbiamo utilizzato l'insieme *MuseiGrandi* definito in precedenza)

- Per ogni biglietto combo di tipo 2, i musei a cui è garantito l'accesso sono 2, uno appartiene all'insieme dei musei di tipo grande e l'altro appartiene all'insieme dei musei di tipo piccolo. Gli attributi possono essere NULL se l'utente non ha ancora scelto in quale museo entrare.

$\forall bc \in \text{BigliettiCombo}$ per cui $\exists b \in \text{Biglietti}$ tale che
 tale che $bc.IDBiglietto = b.IDBiglietto \wedge bc.TipoCombo = 2 \Leftrightarrow bc.IDMuseo1 \in \text{MuseiGrandi} \wedge bc.IDMuseo2 \in \text{MuseiPiccoli}$

(Abbiamo utilizzato gli insiemi *MuseiGrandi* e *MuseiPiccoli* definiti in precedenza)

- Per ogni biglietto di tipo singolo, combo, e abbonamento settimanale, il periodo di validità è di una settimana.

Definizione insieme ausiliario:

$\forall a \in \text{Abbonamenti} \exists b \in \text{Biglietti}$ tale che
 tale che $\text{AbbonamentiSettimanali} = \{b.IDBiglietto \text{ tale che } b.IDBiglietto = a.IDBiglietto \wedge b.Durata = 7\}$

Vincolo:

$\forall bs \in \text{BigliettiSingoli}, bc \in \text{BigliettiCombo}, as \in \text{AbbonamentiSettimanali}$ per cui
 per cui $\exists b \in \text{Biglietti}$ tale che $bs.IDBiglietto = b.IDBiglietto \vee$
 $\vee bc.IDBiglietto = b.IDBiglietto \vee as.IDBiglietto = b.IDBiglietto$ (condizione già controllata nella definizione dell'insieme) \Leftrightarrow
 $\Leftrightarrow b.Durata = 7$

4. Interrogazioni in SQL

a. Proiezione, join e restrizione

Stampa IDUtente e comune di residenza degli utenti che risiedono presso il Comune di Pisa.

```
SELECT (U.IDUtente, R.ComuneResidenza)

FROM (Utenti U, Recapiti R)

WHERE ( U.IDUtente = R.IDUtente AND R.ComuneResidenza = "PISA")
```

b. Group by con having, where e sort

Stampa nome e cognome e numero di biglietti attivi di tutti gli utenti che posseggono almeno 3 biglietti di costo maggiore di €5 (ovvero, biglietti non singoli), in ordine crescente

```
SELECT (U.Nome, U.Cognome, COUNT (*) AS NBiglietti)

FROM (Biglietti B, Utenti U)

WHERE (B.IDUtente = U.IDUtente AND B.Prezzo > 5)

GROUP BY (U.IDUtente, U.Nome, U.Cognome)

HAVING (NBiglietti > 3)

ORDER BY (NBiglietti)
```

c. Join, group by con having e where;

Stampa IDUtente e numero dei biglietti attivi di tutti gli utenti che posseggono almeno due biglietti combo di tipo 2 attivi.

```
SELECT (IDUtente, COUNT (*) AS NBiglietti)

FROM (Biglietti B, BigliettiCombo BC, Utenti U)

WHERE (U.IDUtente = B.IDUtente AND B.IDBiglietto = BC.IDBiglietto AND
BC.TipoCombo = "2")

GROUP BY (IDUtente)

HAVING (NBiglietti > 1)
```

d. Select annidata con quantificatore esistenziale

Stampa nome e cognome degli utenti che hanno almeno un abbonamento annuale attivo

```
SELECT (Nome, Cognome)

FROM (Utenti U)

WHERE EXISTS (

SELECT *

FROM (Biglietti B, Abbonamenti A)

WHERE (U.IDUtente = B.IDUtente AND
      B.IDBiglietto = A.IDBiglietto AND B.Durata = 365)

)
```

e. Select annidata con quantificatore universale

Stampa IDUtente di tutti gli utenti che non posseggono alcun biglietto combo attivo.

```
SELECT (IDUtente)

FROM (Utenti U)

WHERE NOT EXISTS (

SELECT *

FROM (Biglietti B, BigliettiCombo BC)

WHERE (U.IDUtente = B.IDUtente AND B.IDBiglietto = BC.IDBiglietto)

)
```

f. Subquery di confronto quantificato usando una subquery

Stampa nome e cognome di tutti gli utenti che hanno almeno un biglietto con scadenza settimanale attivo.

```
SELECT (Nome, Cognome)

FROM (Utenti U)

WHERE (U.IDUtente = ANY (

SELECT (B.IDUtente)

FROM (Biglietti B)

WHERE (B.Durata = 7)

)
```

5. Piani di accesso

Abbiamo deciso di raggruppare i piani di accesso richiesti per query.

- **Query A**

Proiezione, join e restrizione

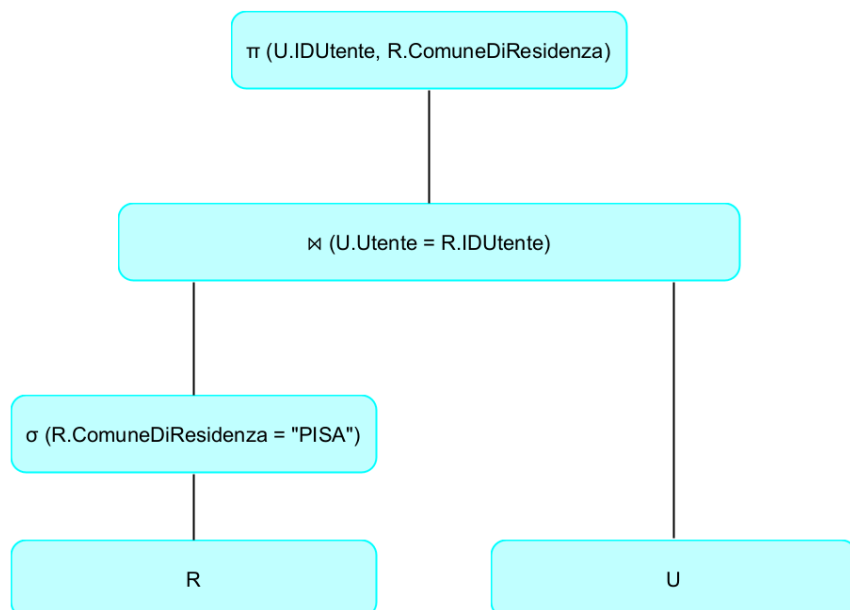
Stampa IDUtente e comune di residenza degli utenti che risiedono presso il Comune di Pisa.

```
SELECT (U.IDUtente, R.ComuneResidenza)

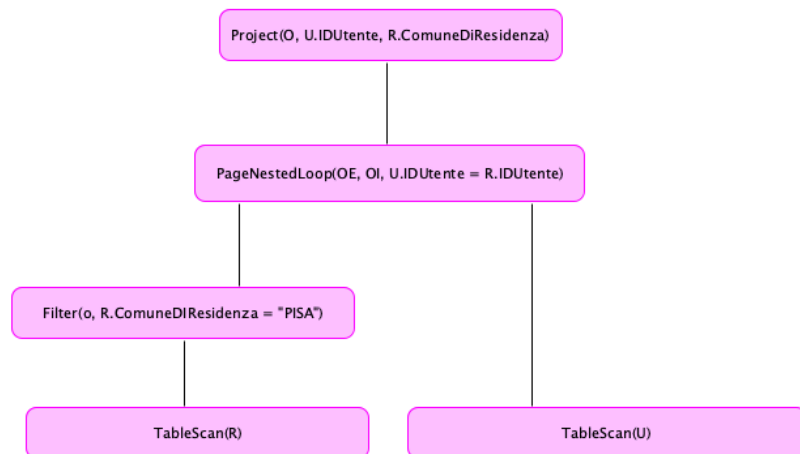
FROM (Utenti U, Recapiti R)

WHERE ( U.IDUtente = R.IDUtente AND R.ComuneResidenza = "PISA")
```

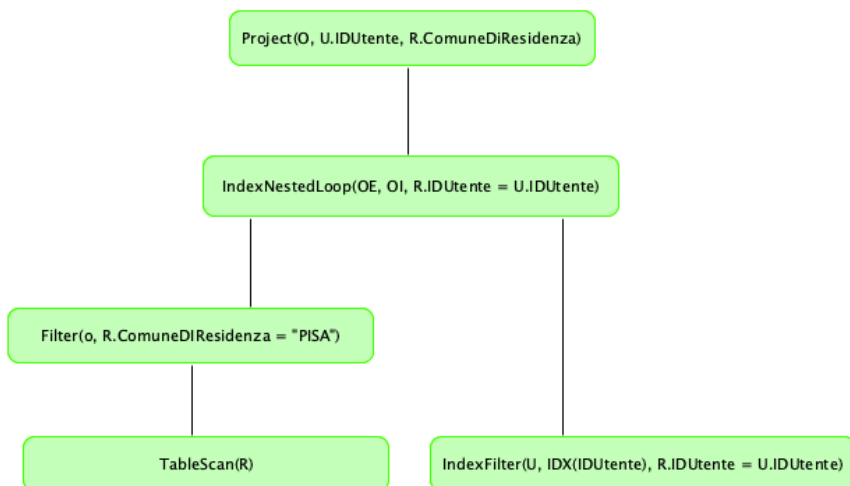
1. Piano di accesso logico



2. Piano di accesso fisico senza indici



3. Piano di accesso fisico con indici



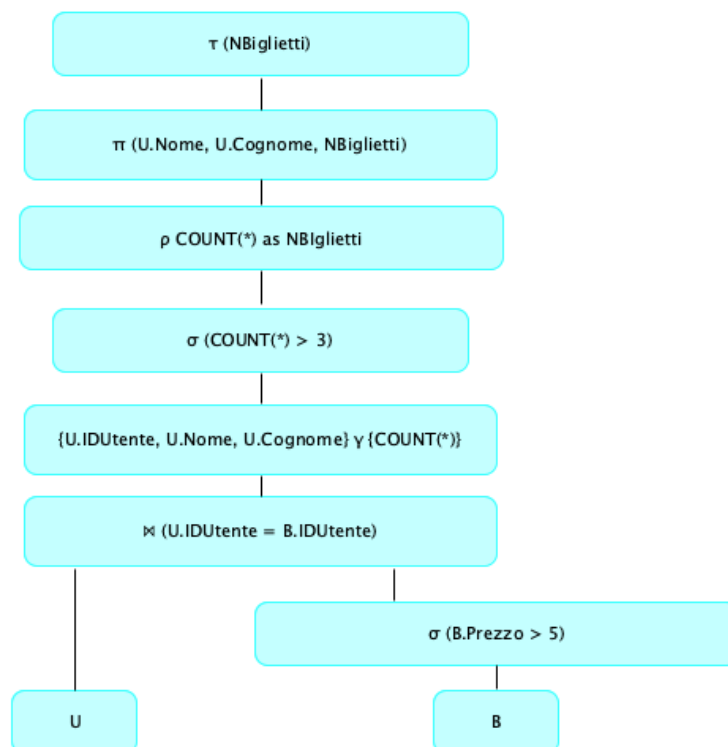
- **Query B**

Group by con having, where e sort

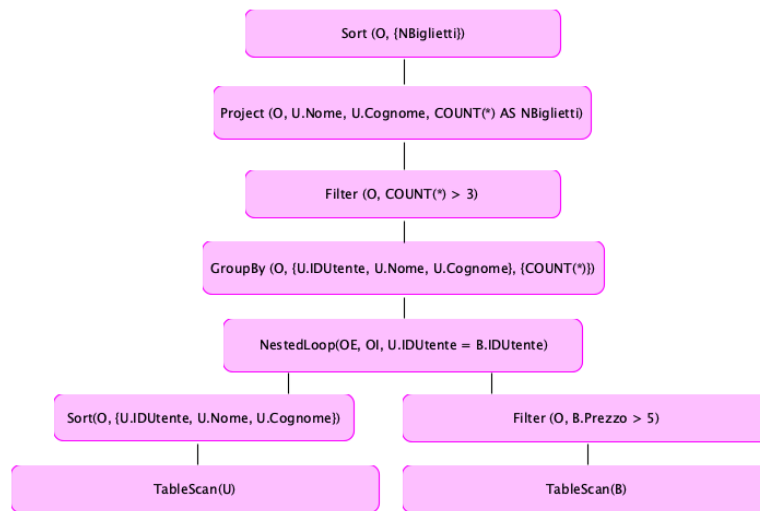
Stampa nome e cognome e numero di biglietti attivi di tutti gli utenti che posseggono almeno 3 biglietti di costo maggiore di €5 (ovvero, biglietti non singoli), in ordine crescente

```
SELECT (U.Nome, U.Cognome, COUNT (*) AS NBiglietti)
FROM (Biglietti B, Utenti U)
WHERE (B.IDUtente = U.IDUtente AND B.Prezzo > 5)
GROUP BY (U.IDUtente, U.Nome, U.Cognome)
HAVING (NBiglietti > 3)
ORDER BY (NBiglietti)
```

1. Piano di accesso logico

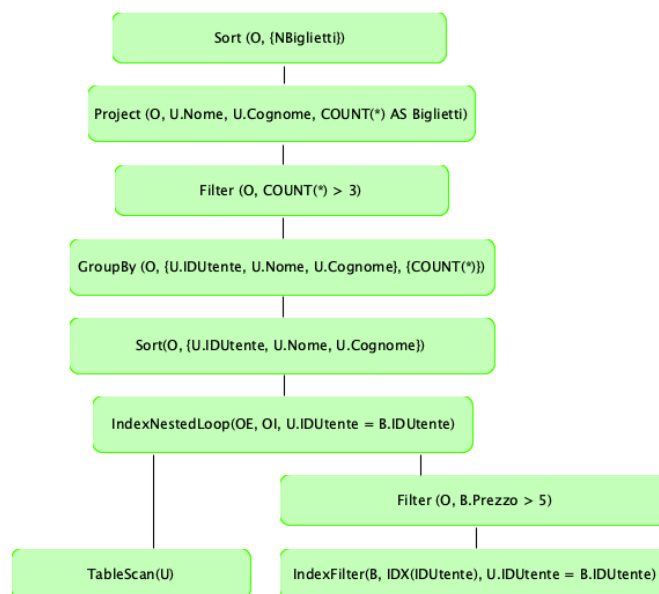


2. Piano di accesso fisico senza indici



Non è stato possibile evitare l'operazione **Sort**, tuttavia abbiamo deciso di anticiparla e sfruttare la proprietà di ordinamento del **NestedLoop**, che ordina le n-uple sugli attributi di giunzione della tabella esterna.

3. Piano di accesso fisico con indici



Neanche in questo caso è stato possibile evitare l'operazione di **Sort** in quanto il raggruppamento viene eseguito sugli attributi **IDUtente**, **Nome** e **Cognome**, ma l'**IndexFilter** è stato effettuato solo sull'attributo **IDUtente**, pertanto l'**IndexNestedLoop** senza **Sort** non avrebbe garantito l'ordinamento richiesto dalla **GroupBy**.

- **Query C**

Join, group by con having e where;

Stampa IDUtente e numero dei biglietti attivi di tutti gli utenti che posseggono almeno due biglietti combo di tipo 2 attivi.

```
SELECT (IDUtente, COUNT (*) AS NBiglietti)

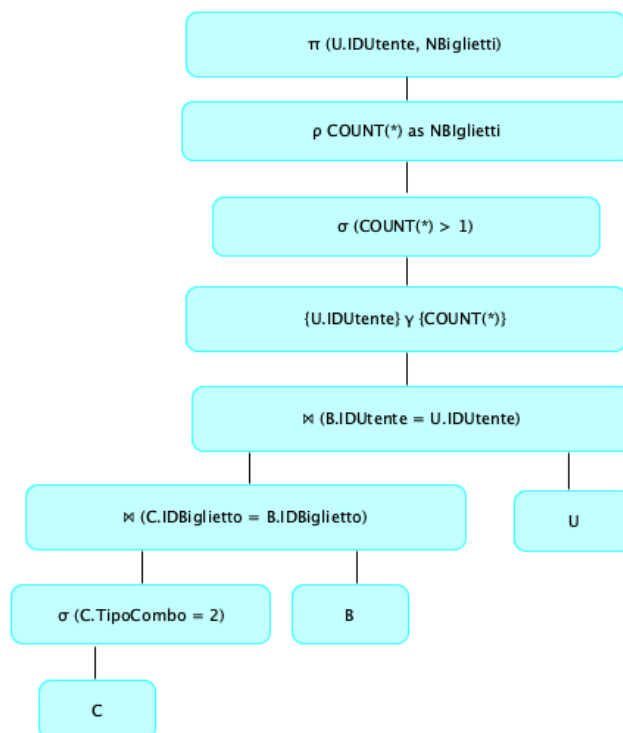
FROM (Biglietti B, BigliettiCombo BC, Utenti U)

WHERE (U.IDUtente = B.IDUtente AND B.IDBiglietto = BC.IDBiglietto AND
BC.TipoCombo = "2")

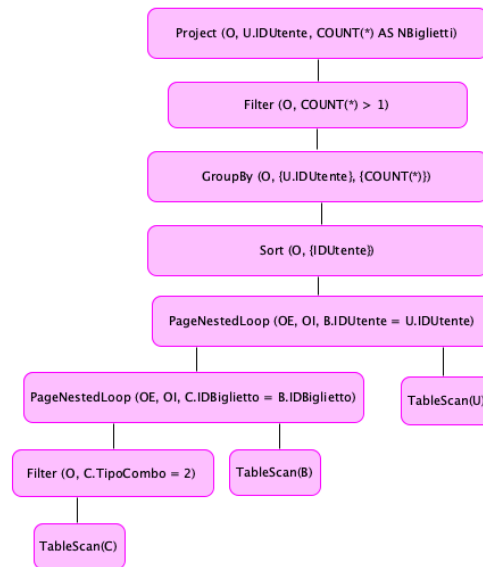
GROUP BY (IDUtente)

HAVING (NBiglietti > 1)
```

1. Piano di accesso logico

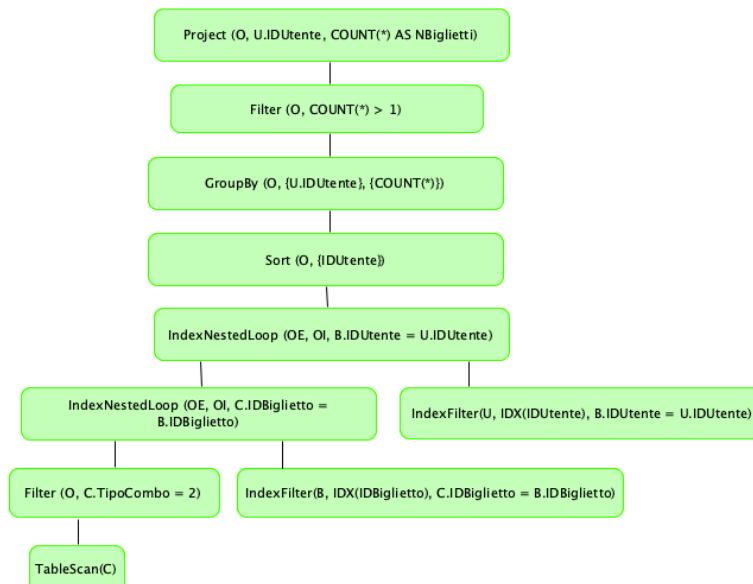


2. Piano di accesso fisico senza indici



L'operazione di Sort non può essere evitata perché l'operazione PageNestedLoop (che abbiamo scelto di usare poiché la tabella esterna è tendenzialmente più piccola di quella interna in entrambe le giunzioni), mantiene l'ordinamento per pagine, cosa non utile alla GroupBy successiva.

3. Piano di accesso fisico con indici



Il primo IndexNestedLoop ordina la tabella risultante dalla giunzione per IDBiglietto. IDUtente non è sicuramente chiave di questa tabella risultante quindi la giunzione risultante dal secondo IndexNestedLoop con indice IDutente, non sarà ordinata per l'attributo richiesto dalla GroupBy.