

ESERCIZIO 1

DOMINIO:

Ricetta = Una ricetta può essere letta da più Birrai (che hanno la relativa autorizzazione) ma può essere scritta/modificata solo da un birraio. In una ricetta vi sono vari ingredienti (con relative quantità e facoltatività) e per ogni ricetta possono esistere molteplici lotti

Birraio = Un birraio può leggere più ricette(in base all'autorizzazione che possiede) e può crearne varie. Ad un birraio sono associati più fornitori, più strumenti e più lotti

Fornitore = Un fornitore può produrre più ingredienti e rifornire più birrai

Cliente = Un cliente può effettuare diversi ordini

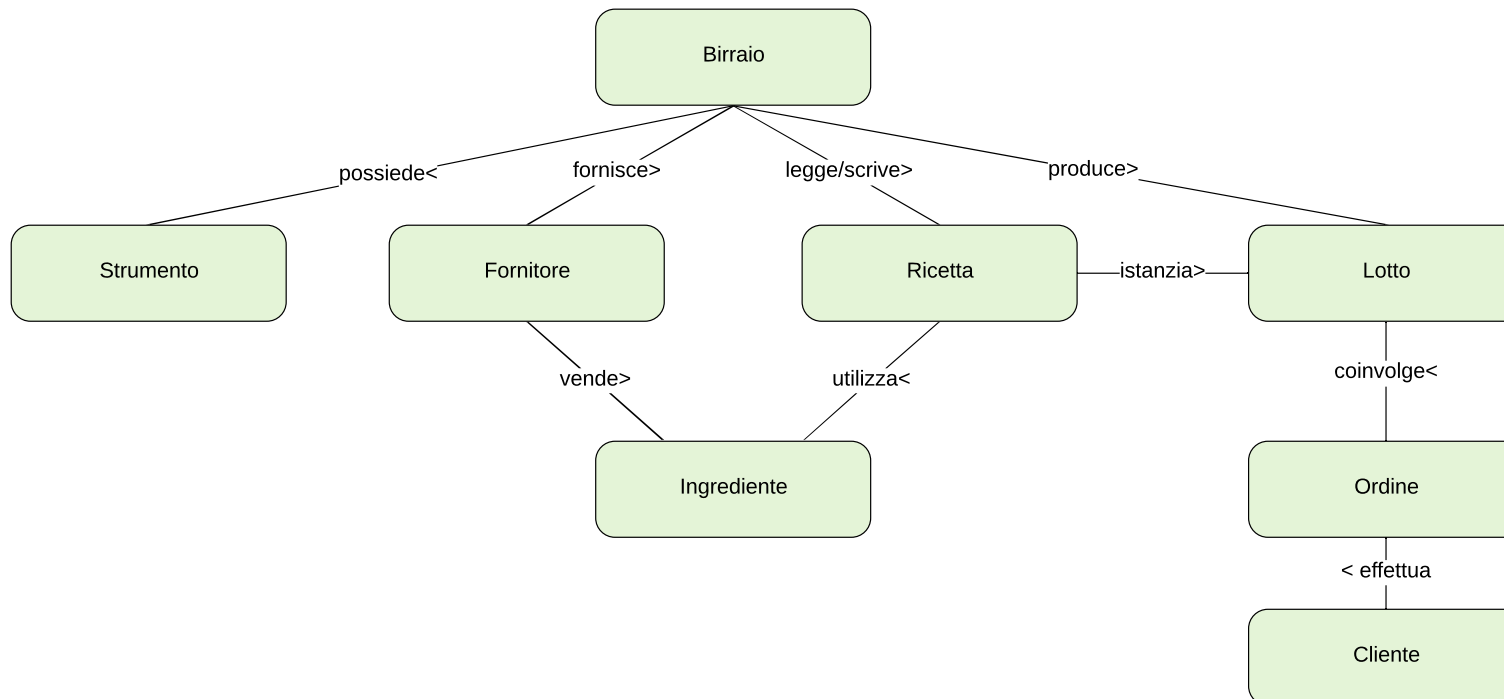
Lotto = Un lotto può essere prodotto da un solo birraio e appartiene ad una sola ricetta ma può appartenere a diversi ordini.

Ingrediente= Un ingrediente può essere presente in diverse ricette in quantità diversa e con una diversa facoltatività. Può essere prodotto da diversi fornitori

Ordini = Un ordine può essere effettuato da un singolo cliente e su un singolo lotto

Strumento= Uno strumento appartiene ad un singolo birraio

Non avendo creato delle sottoclassi non abbiamo una vera e propria gerarchia. Abbiamo però ideato una pseudo gerarchia che idealmente parte da birraio e si dirama nelle altre classi. Considero livello di gerarchia il numero minimo di associazioni che io attraverso per arrivare dal birraio alla classe scelta.



PSEUDOGERARCHIA:

Lvl.0:

Birraio

Lvl.1:

Ricetta

Fornitore

Strumento

Lotto

Lvl.2:

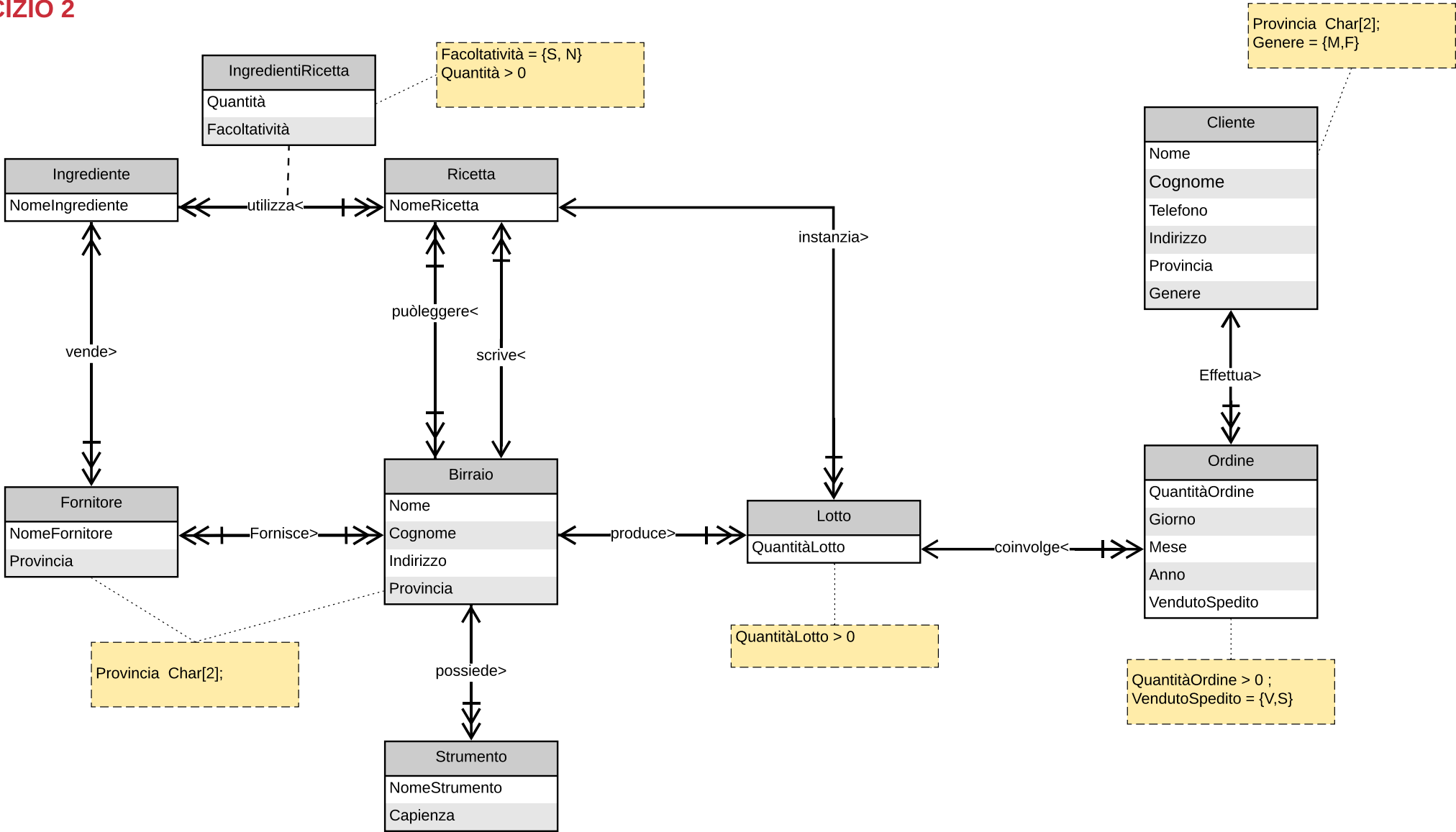
Ingrediente

Ordine

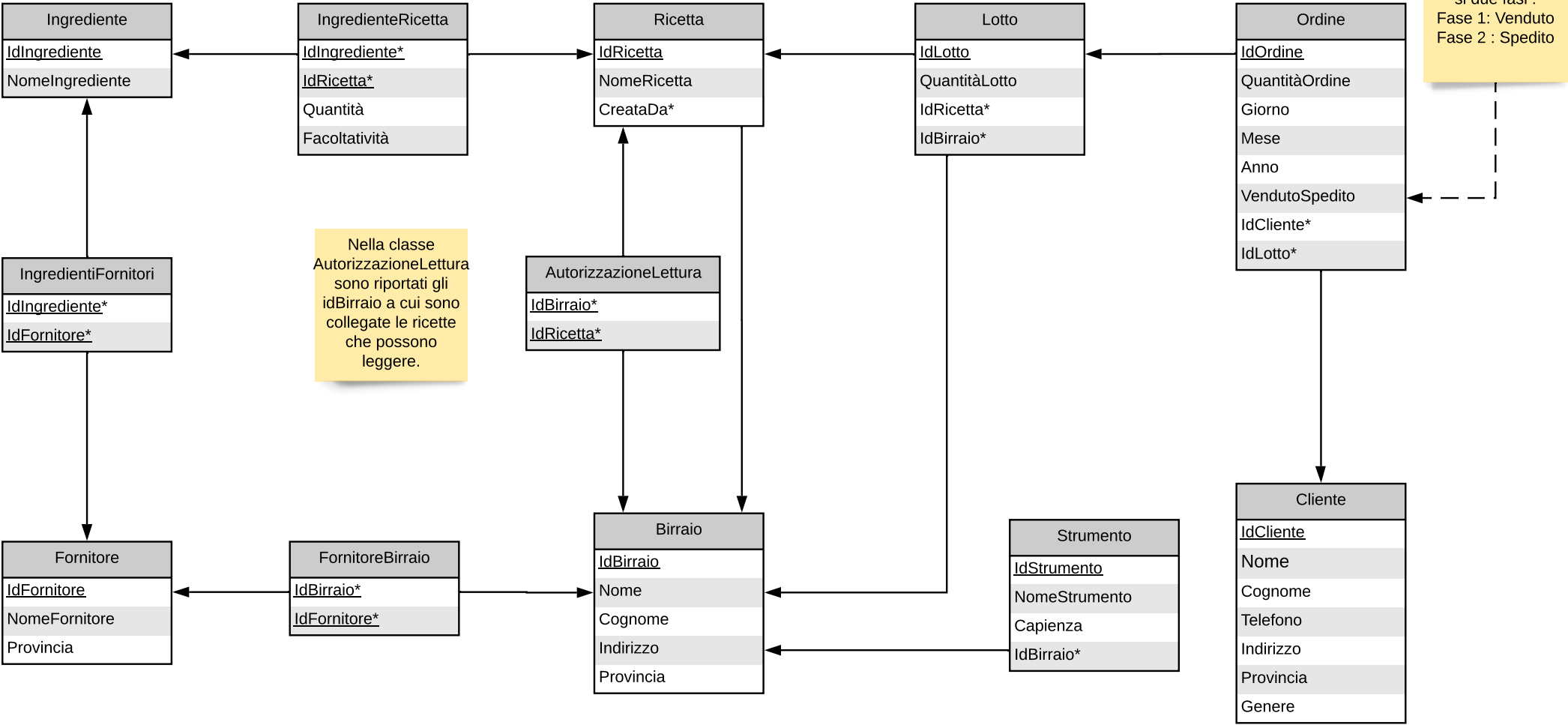
Lvl.3:

Cliente

ESERCIZIO 2



ESERCIZIO 3



Ricetta(IdRicetta, NomeRicetta, CreatoDa*)

Birraio(IdBirraio, Nome, Cognome, IndirizzoBirraio, Provincia)

FornitoreBirraio(IdBirraio*, IdFornitore*)

Fornitore(IdFornitore, NomeFornitore, Provincia)

IngredienteRicetta(IdIngrediente*, IdRicetta*, Quantità, Facoltatività)

Ingrediente(IdIngrediente, NomeIngrediente)

Ordine(idOrdine, QuantitàOrdine, Giorno, Mese, Anno, VendutoSpedito, IdCliente*, IdLotto*)

Lotto(IdLotto, QuantitàLotto, IdBirraio*, IdRicetta*)

Cliente(IdCliente, Nome, Cognome, Indirizzo, Telefono, Provincia, Genere);

Strumento(IdStrumento, NomeStrumento, Capienza, IdBirraio*)

IngredientiFornitori(IdIngrediente*, IdFornitore*)

DIPENDENZA FUNZIONALI:

Birraio(IdB -> Nome, Cognome, Indirizzo, Provincia)

Cliente(idC -> Nome, Cognome, Telefono, Indirizzo, Provincia, Genere)

Fornitore(IdF->NomeFornitore Provincia)

Ricetta(IdR -> NomeRicetta, CreataDa)

Ingrediente(IdI->NomeIngrediente)

Strumento(IdS->NomeStruemento, Capienza, IdBirraio)

Ordine(IdO -> QuantitàOrdine, Giorno, Mese, Anno, VendutoSpedito, IdC, IdL)

Lotto(IdL -> QuantitàLotto, IdB, IdR)

AutorizzazioniLettura()

IngredienteRicetta()

FornitoreBirraio()

IngredientiFornitori()

LE DIPENDENZE SONO TUTTE IN FNBC

ESECIZIO 4

a) Riportare Nome e Cognome dei Birrai che hanno tra gli strumenti 'Filtro Bazooka'

```
SELECT DISTINCT b.Nome, b.Cognome
FROM Birraio b JOIN Strumenti s ON b.IdBirraio = s.IdBirraio
WHERE s.NomeStrumento = 'Filtro Bazooka';
```

b) Riportare la somma e il relativo anno in cui i litri di birra totali ordinati sono maggiori di 100, negli anni compresi tra 2016 e 2019

```
SELECT o.Anno, SUM(o.QuantitàOrdine) as LitriDiBirraOrdinati
FROM Ordine o
WHERE 2016 < o.Anno < 2019
GROUP BY o.Anno
HAVING SUM(o.QuantitàOrdine) > 100
ORDER BY o.Anno
```

c) Riportare Nome e Cognome di ogni cliente che ha provincia = 'PI' e non ha ordini incompleti (tutti gli ordini associati sono in stato "Spedito")

```
SELECT c.Nome, c.Cognome
FROM Clienti c JOIN Ordini o ON c.IdCliente = o.IdCliente
WHERE c.Provincia = 'PI'
GROUP BY c.IdCliente
HAVING Max(VendutoSpedito) = "S";
```

d) Riportare IdBirraio, Nome e Cognome dei birrai che hanno prodotto almeno un lotto da 30 litri

```
SELECT b.IdBirraio, b.Nome, b.Cognome
FROM Birraio b
WHERE EXISTS (SELECT *
              FROM Lotti l
              WHERE b.IdBirraio = l.IdBirraio AND l.QuantitàLotto >= 30);
```

e) Per ogni Fornitore che rifornisce solo birrai con provincia = 'MI', Riportare IdFornitore, NomeFornitore e Provincia

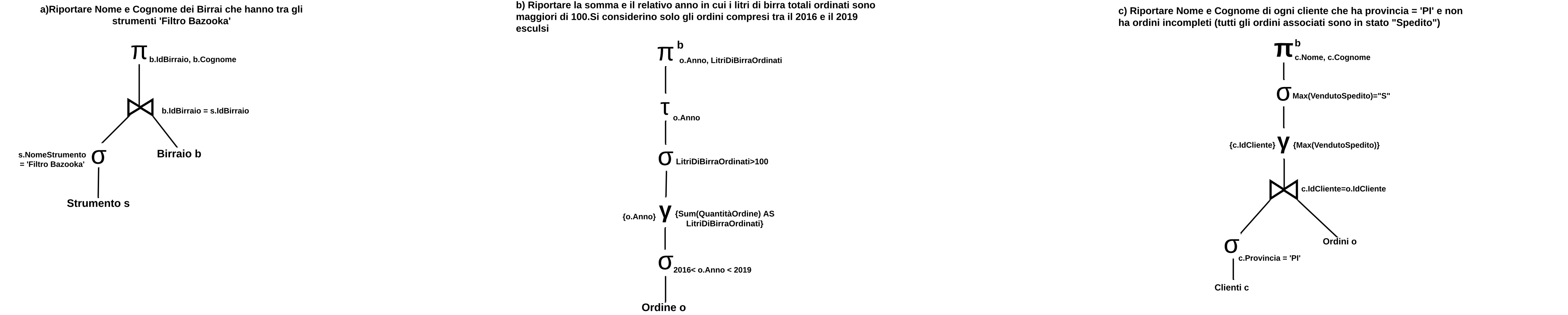
```
SELECT f.IdFornitore, f.NomeFornitore, f.Provincia
FROM Fornitore f
WHERE NOT EXISTS (SELECT *
                  FROM FornitoreBirraio JOIN Birraio b ON b.IdBirraio = fb.IdBirraio
                  WHERE f.IdFornitore = fb.IdFornitore
                  AND NOT(b.Provincia = 'MI'));
```

f) Riportare IdRicetta e Nome delle ricette per le quali i lotti associati producono tutti almeno 40 litri di birra. Inoltre riportare IdBirraio e Cognome del Birraio che le ha scritte

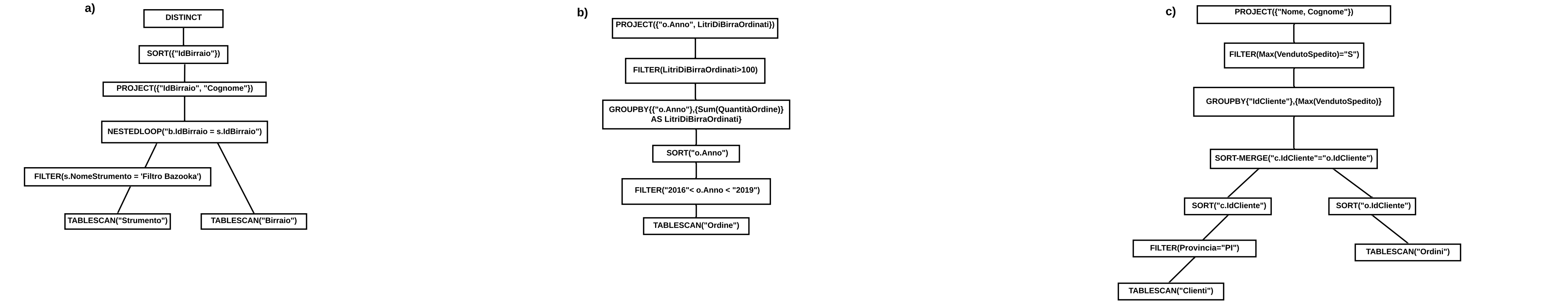
```
SELECT r.IdRicetta, r.NomeRicetta, b.IdBirraio, b.Cognome
FROM Ricetta r JOIN Birraio b ON r.CreatoDa = b.IdBirraio
WHERE 40 <= ALL (SELECT l.QuantitàLotto
                FROM Lotti l
                WHERE r.IdRicetta = l.IdRicetta);
```

ESERCIZIO 5

a)Alberi Logici



b)Alberi Fisici(senza Indici)



c)Alberi Fisici(con Indici)

