

# Interrogazioni SQL

P. Rullo

## Schema della base di dati

- **Birra** (nome, marca, colore, gradi)
- **Bar** (codBar, nome, città)
- **Bevitore** (codBev, nome, età, città)
- **Piace** (bev\*, birra\*)
- **Vende** (bar\*, birra\*, prezzo, quant)

## Selezione-proiezione-join

1. Codice dei bar che vendono birre rosse
2. Nome dei bar che vendono birre rosse ad un prezzo minore di 5€
3. Nome ed età dei bevitori a cui piacciono birre con gradazione > 5
4. Coppie di bevitori per cui esiste qualche birra che piace ad entrambi
5. Coppie di bevitori omonimi e coetanei

## Unione-Intersezione

6. Bevitori che sono di Roma o a cui piace la Nastro Azzurro
7. Codici di birre che piacciono sia al bevitore b1 che b2

## Negazione – Not In, Not Exists, Minus

8. Codici dei bar il cui nome è unico
9. Codici dei bar che vendono la Nastro Azzurro al prezzo minimo
10. Bar che vendono tutte le birre
11. Bar che vendono tutte le birre vendute dal bar B2
12. Bevitori che bevono solo le birre vendute dal bar B1
13. Bevitori che bevono tutte e solo le birre bevute dal bevitore X
14. Età minima tra tutti i bevitori della birra Nastro Azzurro

## Funzioni di Aggregazione – Groupby - Having

15. Numero bevitori a cui piace la birra Nastro Azzurro
16. Numero ed età media dei bevitori a cui piacciono birre con gradi >6
17. Per ogni birra, prezzo minimo a cui viene venduta
18. Per ogni birra, prezzo minimo e bar che la vendono
19. codici bar che vendono una quantità totale di birra > 100
20. codici bar che vendono una quantità totale di birra > di quella venduta dal bar B1
21. codici bar che vendono più birre di quante vendute complessivamente da B1 e B2

## Soluzioni proposte

### Selezione-proiezione-join

3. Nome ed età dei bevitori a cui piacciono birre con gradazione > 5

```
Select Be.nome, età
From Bevitore ad Be, Piace as P, Birra as Bi
Where Be.codBev=P.bev and P.birra=Bi.nome and grad>5
```

4. Coppie di bevitori per cui esiste qualche birra che piace ad entrambi

```
Select X.bev, Y.bev
From Piace ad X, Piace as Y
Where X.bev <> Y.bev and X.birra=Y.birra
```

### Unione-Intersezione

7. Codici di birre che piacciono sia al bevitore b1 che b2

```
Select birra
From Piace as P
Where bev=p1
      INTERSECT
Select birra
From Piace as P
Where bev=p2
```

### Negazione

8. Codici dei bar il cui nome è unico

```
Select codBar
From Bar as B1
Where NOT EXISTS
      (Select *
      From Bar as B2
      Where B1.codBar <> B2.codBar and B1.nome=B2.nome)
```

### Formulazione alternativa

```
Select codBar
From Bar as B1
Where codeBar NOT IN
      (Select codeBar
      From Bar as B2
      Where B1.codBar <> B2.codBar)
```

9. Codici dei bar che vendono la Nastro Azzurro al prezzo minimo

```
Select bar
From Vende V1
Where birra= nastro azzurro and NOT EXISTS
    (Select *
     From Vende as V2
     Where birra= nastro azzurro and V2.prezzo<V1.prezzo)
```

Formulazione alternativa:

```
Select bar
From Vende
Where birra= nastro azzurro and prezzo =
    (Select Min(Prezzo)
     From Vende
     Where birra= nastro azzurro)
```

10. Bar che vendono tutte le birre

```
Select codBar
From Bar as X
Where NOT EXISTS
    (Select *
     From Birra as Y
     Where NOT EXISTS
        (Select *
         From Vende as V
         Where V.bar=X.bar and V.birra=Y.nome))
```

13. Bevitori che bevono tutte e solo le birre bevute dal bevitore X

```
Select * //TUTTE
From Bev as B
Where NOT EXISTS
    (select *
     From Piaci as P1
     Where P1.bev=X and NOT EXISTS
        (select *
         From Piaci as P2
         Where P2.bev=B.codBev and P2.birra=P1.birra))
```

INTERSECT

```
Select * //SOLO
From Bev as B
Where NOT EXISTS
    (select *
     From Piaci as P1
```

```

Where P1.bev=B.bev and NOT EXISTS
(select *
From Piace as P2
Where P2.bev=X and P2.birra=P1.birra)

```

14. Età minima tra tutti i bevitori della birra Nastro Azzurro

```

Select min(età)
From bevitore as B, piace as P
Where B.codBev=P.bev and P.birra=nastro azzurro

```

Formulazione alternativa

```

Select età
From bevitore as B, piace as P
Where B.codBev=P.bev and P.birra=nastro Azzurro and età <=ALL
    (Select età
     From bevitore as B, piace as P
     Where B.codBev=P.bev and P.birra=nastro Azzurro)

```

18. Per ogni birra, prezzo minimo e bar che la vendono

```

Select bar, birra, prezzo
From Vende
Where (birra, prezzo) IN
    (Select birra, min(prezzo)
     From Vende
     groupBy birra)

```

Formulazione alternativa

```

Select bar, birra, prezzo
From Vende as V1
Where Prezzo = (Select min(prezzo)
                From Vende as V2
                Where V2.birra=V1.birra)

```

Formulazione alternativa

```

Define view Birra_prezzoMin(birra, prezzoMin) As
(Select birra, min(prezzo)
 From Vende
 groupBy birra)

```

```

Select bar, birra, prezzo
From Vende as V, Birra_prezzoMin B
Where V.birra=B.birra and V.prezzo=B.prezzoMin

```

## Funzioni di Aggregazione – Groupby - Having

21. codici bar che vendono più birre di quante vendute complessivamente da B1 e B2

```
Select bar
From Vende
groupBy bar
having count(*) > (Select count(*)
                  From Vende
                  Where bar=B1 or bar=B2)
```