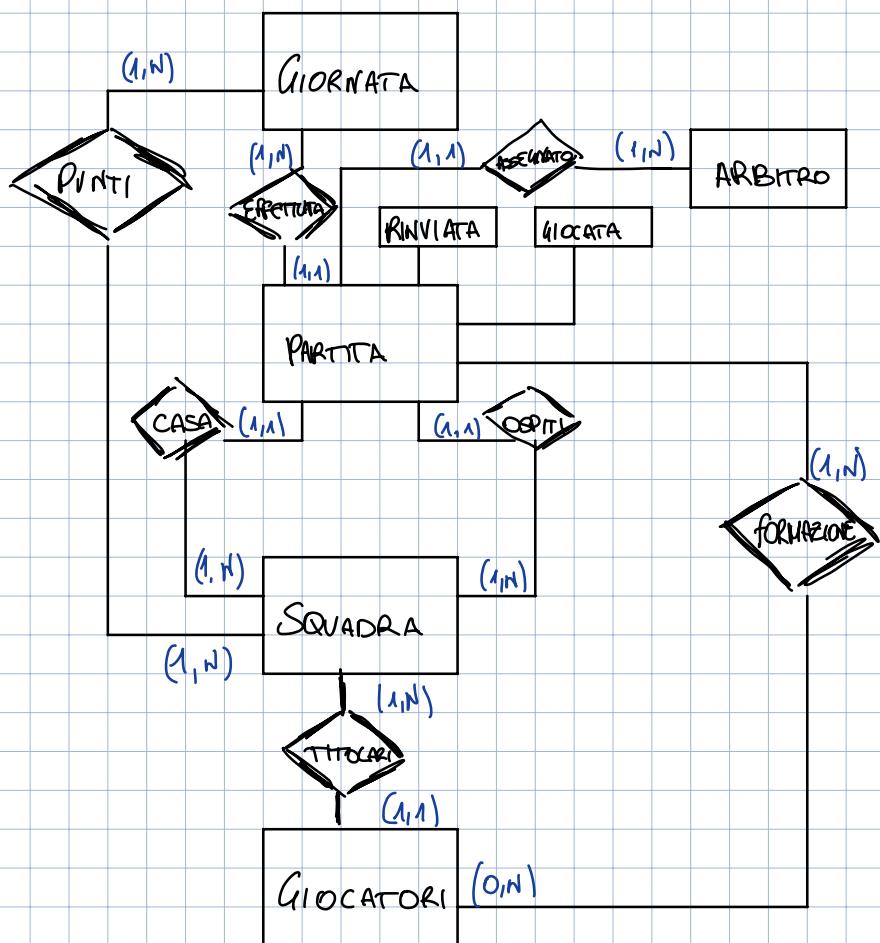
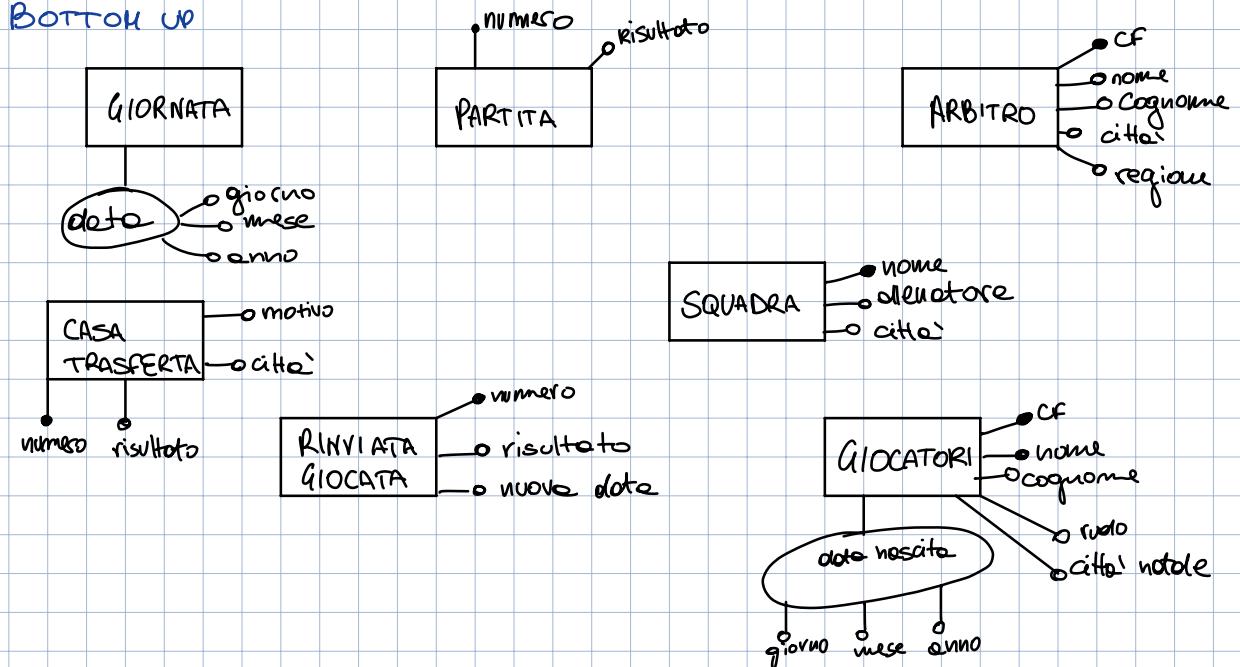
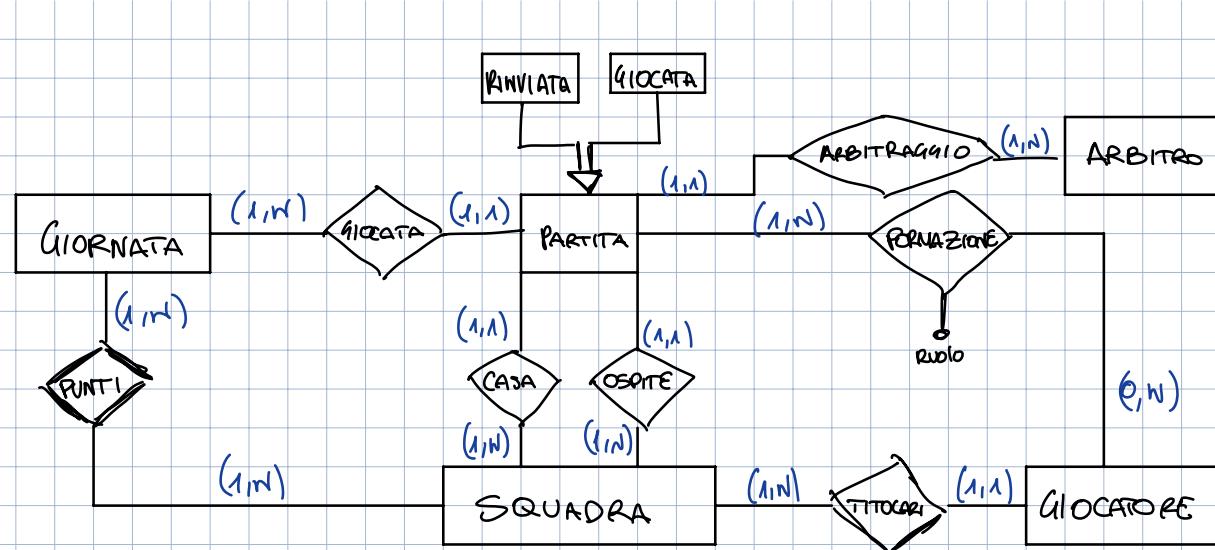


TOP DOWN



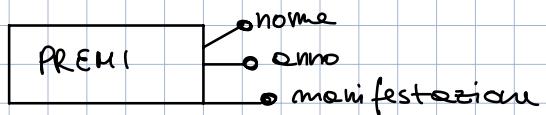
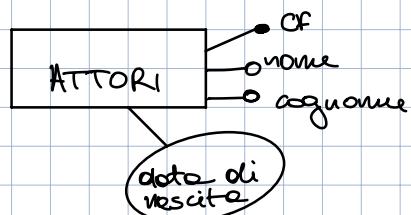
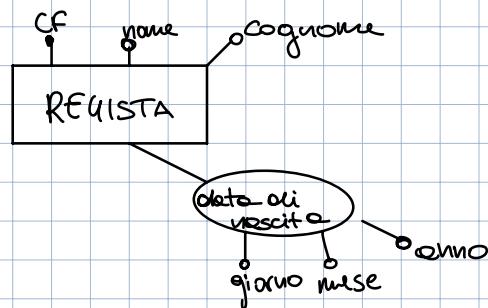
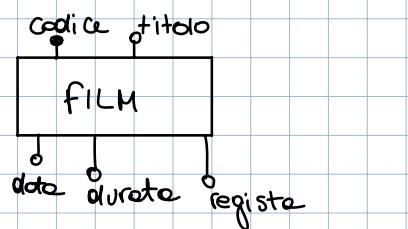
BOTTOM UP

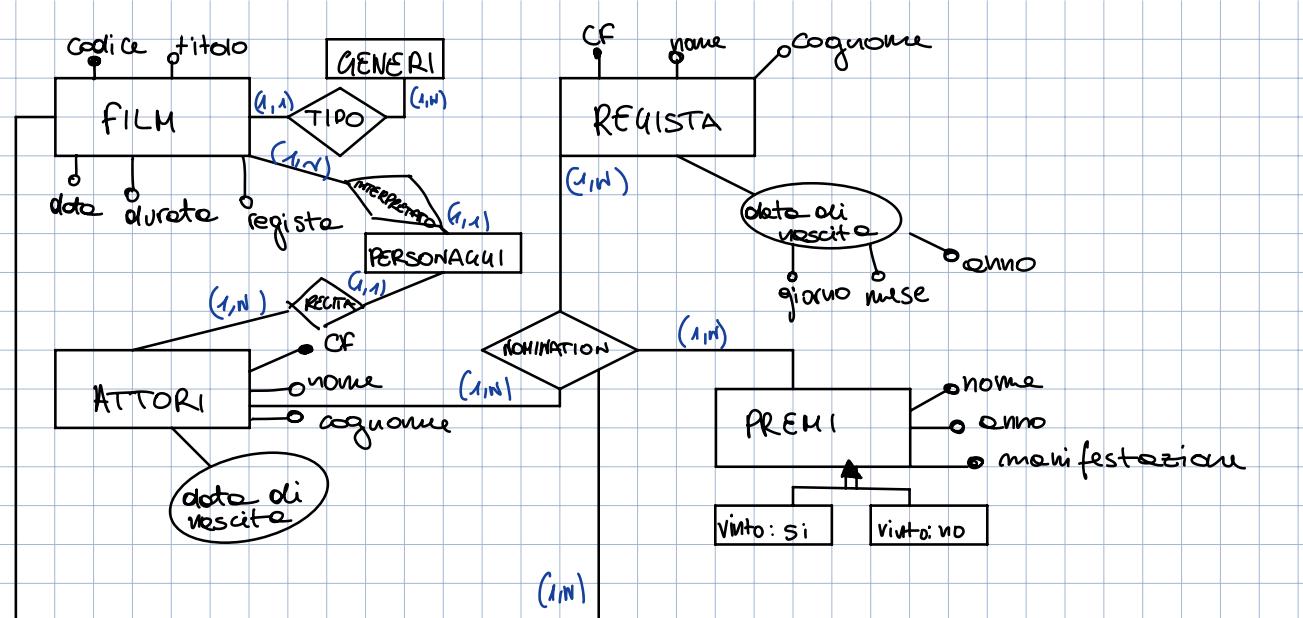




2° APPELLO COMPLETO

BOTTOM-UP





Film (codice, titolo, genere, durata, data, regista*)

Attori (CF, nome, cognome, data di nascita)

Regista (CF, nome, cognome, data di nascita)

Premi (nome, anno, manifestazione, vincitore)

Regista . CF REFERENCES Film . regista

Personaggi (codafilm, titolo, attore)

QUERY:

1-

```

SELECT u.idUtente, u.Email, p.idUtente
FROM Utente AS u JOIN Previsione AS p ON u.idUtente = p.idUtente
WHERE p.Umidita' = (SELECT min(Umidita')
                    FROM Previsione)
    
```

2-

```

SELECT c.nome, c.Regione, c.Stato
FROM Città AS c JOIN StazioniMeteo AS s ON s.città = c.nome
    
```

2 APPELLO SQL

1-

```
SELECT IdUtente, Email  
FROM Utente AS U JOIN Previsione AS P ON U.IdUtente = P.IdUtente  
WHERE P.Umidita = (SELECT min(Umidita)  
                    FROM Previsione);
```

2 -

```
SELECT C.Città, C.Regione, AVG(P.Temperatura) AS TempMedia  
FROM Città AS C JOIN StazioneMeteo AS S ON C.Nome = S.città  
JOIN Previsione AS P ON S.IdStazione = P.IdStazione  
GROUP BY C.Città, C.Regione  
HAVING avg(P.Temperatura) = (SELECT max(TempMedia)  
                               FROM (SELECT avg(Temperatura) AS TempMedia  
                                     FROM Città AS C JOIN StazioneMeteo  
                                     AS S ON C.Nome = S.città JOIN Previsione  
                                     AS P ON S.IdStazione = P.IdStazione  
                                     GROUP BY C.Nome) AS TempCittà);
```

3 -

Create view Stazioni

```
SELECT Città, COUNT(*) AS NumStazioni  
FROM StazioneMeteo  
GROUP BY città
```

```
SELECT C.Regione, min(P.Temperatura), max(P.Umidità)  
FROM Città AS C JOIN StazioneMeteo AS S ON C.Nome = S.città  
JOIN Previsione AS P ON S.IdPrevisione = P.IdPrevisione  
GROUP BY Regione  
HAVING NumStazioni >= 1
```

4 -

Create view Stazioni

```
SELECT città, COUNT(*) AS NumStazioni  
FROM StazioneMeteo  
GROUP BY C.Nome
```

```
SELECT C.Provincia, avg(P.Umidità)  
FROM Città AS C JOIN StazioneMeteo AS S ON C.Nome = S.città  
JOIN Previsione AS P ON S.IdStazione = P.IdStazione  
GROUP BY C.Provincia  
HAVING NumStazioni >= 1
```

3° APPELLO

1-

```
SELECT e.IdEmm, e.Nome, t.IdTr (*) COUNT AS NumTransmissioni  
FROM Emettente AS e JOIN Transmissione AS t ON e.IdEmm = t.IdTr  
GROUP BY e.Nome  
HAVING t.IdTr (*) COUNT > (SELECT avg(IdTr)  
                           FROM Transmissioni)
```

2 -

Create view NumTransmissioni
SELECT IdTr (*) COUNT AS Conteggio
FROM Transmissioni

Create view Let_it_be AS

```
SELECT Titolo
```

```
FROM Brano
```

```
WHERE Titolo LIKE = 'Let it be'
```

```
SELECT d.Nome, d.Cognome, t.IdTr, b.Titolo  
FROM D3 AS d JOIN Transmissione AS t ON d.IdDj = t.IdDj  
WHERE Conteggio = 2 AND Let_it_be = True  
GROUP BY d.Nome, d.Cognome
```

ES SQL n° 1

1-

```
SELECT first_name AS Nome, last_name AS Cognome  
FROM employees
```

2-

```
SELECT DISTINCT department_id  
FROM departments
```

3 -

```
SELECT first_name, last_name, salary, salary * 0.15 AS provvident_fund  
FROM employees
```

4 -

```
SELECT first_name, last_name, salary  
FROM employees  
WHERE salary < 10000 OR salary > 15000
```

5 -

```
SELECT e.first_name, e.last_name, e.department_id, d.department_name
FROM employees AS e JOIN departments AS d ON e.department_id =
d.department_id
```

ESAMS 9/02/2021

1-

```
SELECT a.Nome, a.Cognome, f.Titolo, COUNT(DISTINCT p.IdCinema)
FROM Attore AS a JOIN Ruolo AS r ON a.CF = r.CFAttore
JOIN Film AS f ON r.TitoloFilm = f.Titolo
JOIN Proiezione AS p ON f.Titolo = p.TitoloFilm
WHERE a.Nome = 'Bruce' AND a.Cognome = 'Willis'
GROUP BY f.Titolo;
```

2-

```
SELECT f.Regista
FROM Film AS f
WHERE f.Anno = 2019
GROUP BY f.Regista
HAVING COUNT(f.Titolo) > (SELECT COUNT(f.Titolo)
                           FROM Film AS f
                           WHERE f.Anno = 2020
                           GROUP BY Regista);
```

3 -

Create View NazAttori As

```
SELECT f.Titolo, a.CF, a.Nazionalita'
FROM Attore AS a JOIN Ruolo AS r ON a.CF = r.CFAttore
JOIN Film AS f ON r.TitoloFilm = f.Titolo
WHERE a.Nazionalita' = 'Spagnolo' OR a.Nazionalita' = 'francese'
GROUP BY f.Titolo;
```

```
SELECT a.CF, a.Nome
FROM Attore
```

```
WHERE Nazionalita' = 'francese' AND a.CF NOT IN (SELECT CF
                                                 FROM NazAttori
                                                 WHERE Nazionalita' =
                                                 'Spagnola');
```

4 -

```
SELECT c.Id, c.Nome, COUNT(*)
FROM Cinema AS c JOIN Proiezione AS p ON c.Id = p.IdCinema
JOIN Film AS f ON p.TitoloFilm = f.Titolo
WHERE f.Regista = 'Woody Allen'
GROUP BY c.Nome
```

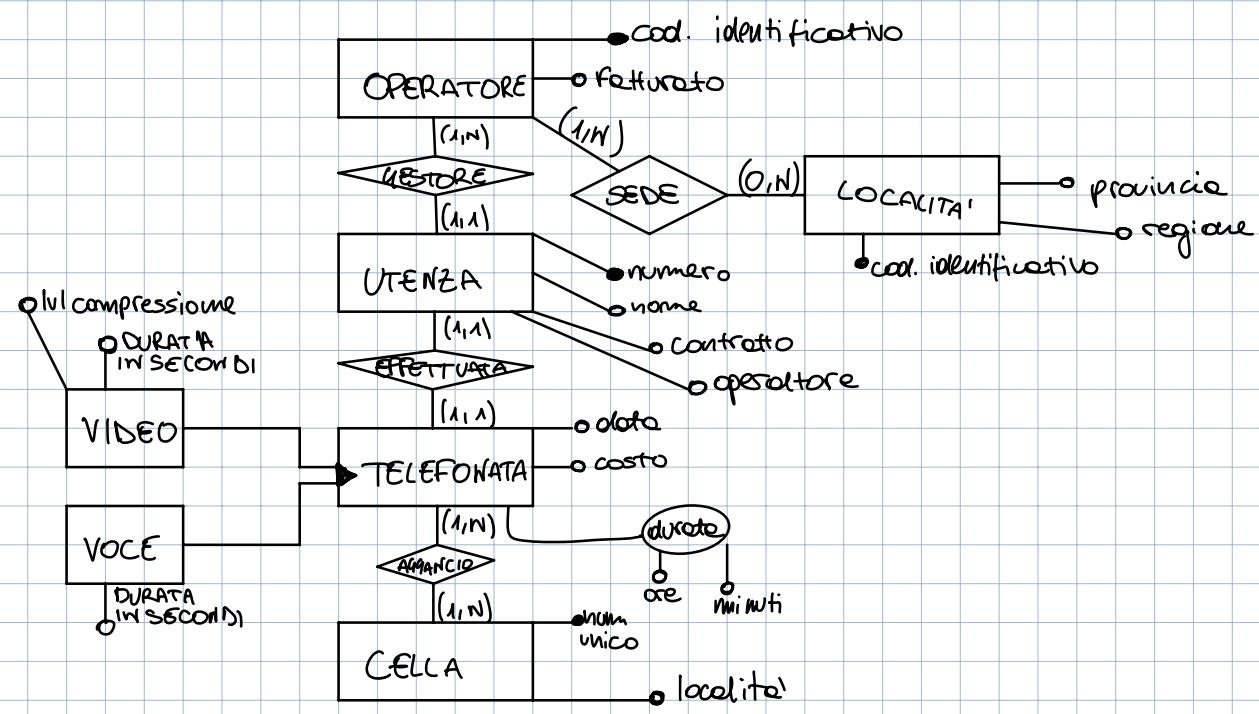
$\text{HAVING COUNT(DISTINCT f.titolo)} = (\text{SELECT COUNT(*)}$
 FROM film
 $\text{WHERE Regista} = 'Woody Allen')$

I APPELLO



TOP DOWN PRIMO PASSACCO

RISULTATO FINALE



Operatore (cod_identificativo, fattureto)

Localita' (cod_identificativo, provincia, regione)

Utente (numero, nome, data, costo, durata)

Telefonata (Utente, durata, costo, tipo, livello_compressione)

Cella (num_unico, localita')

Operatore.CodIdentificativo REFERENCES Utente . Operatore

Telefonata.Utente REFERENCES Utente . Numero

Cella.num_unico REFERENCES Localita'.Cod_identificativo

QUERY

a)

```
SELECT CATEGORIA, AVG(Prezzo), MIN(Prezzo), MAX(Prezzo)
FROM PRODOTTO
GROUP BY CATEGORIA
```

b)

```
Create view Popolare AS (
    SELECT COUNT(IdProdotto), O.data
    FROM Prodotto AS P JOIN Ordine AS O ON P.IdProdotto = O.IdProdotto
    WHERE O.data > '2022-01-01'
    GROUP BY CATEGORIA
    HAVING COUNT(*) > 100);
```

X

```
SELECT U.IdUtente, U.Email, O.data
FROM Utente AS U JOIN Ordine AS O ON U.IdUtente = O.IdUtente
JOIN Prodotto AS P ON O.IdProdotto = P.IdProdotto
WHERE O.data > '2022-01-01'
GROUP BY O.data, U.IdUtente, U.Email
HAVING SUM(O.Quantita) > 100
```

C -

Create view Premium AS (

```
SELECT U.IdUtente, O.Quantita, O.Quantita * prezzo AS costo_tot;
FROM Utente AS U JOIN Ordine AS O ON U.IdUtente = O.IdUtente
WHERE O.Quantita > 10
```

SELECT IdOroline, IdProdotto

d-

SELECT p.IdProd, p.Nome, p.Prezzo
FROM Prodotto AS p JOIN Oroline AS o ON p.IdProd = o.IdProd
JOIN Utente AS u ON o.IdUtente = u.IdUtente
WHERE u.Incivizzo = 'Genova'

GROUP BY p.IdProd

HAVING COUNT(DISTINCT u.IdUtente) = (SELECT COUNT(*)
FROM Utente
WHERE Incivizzo = 'Genova'));

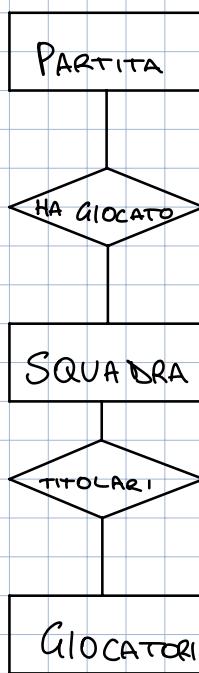
ALGEBRA

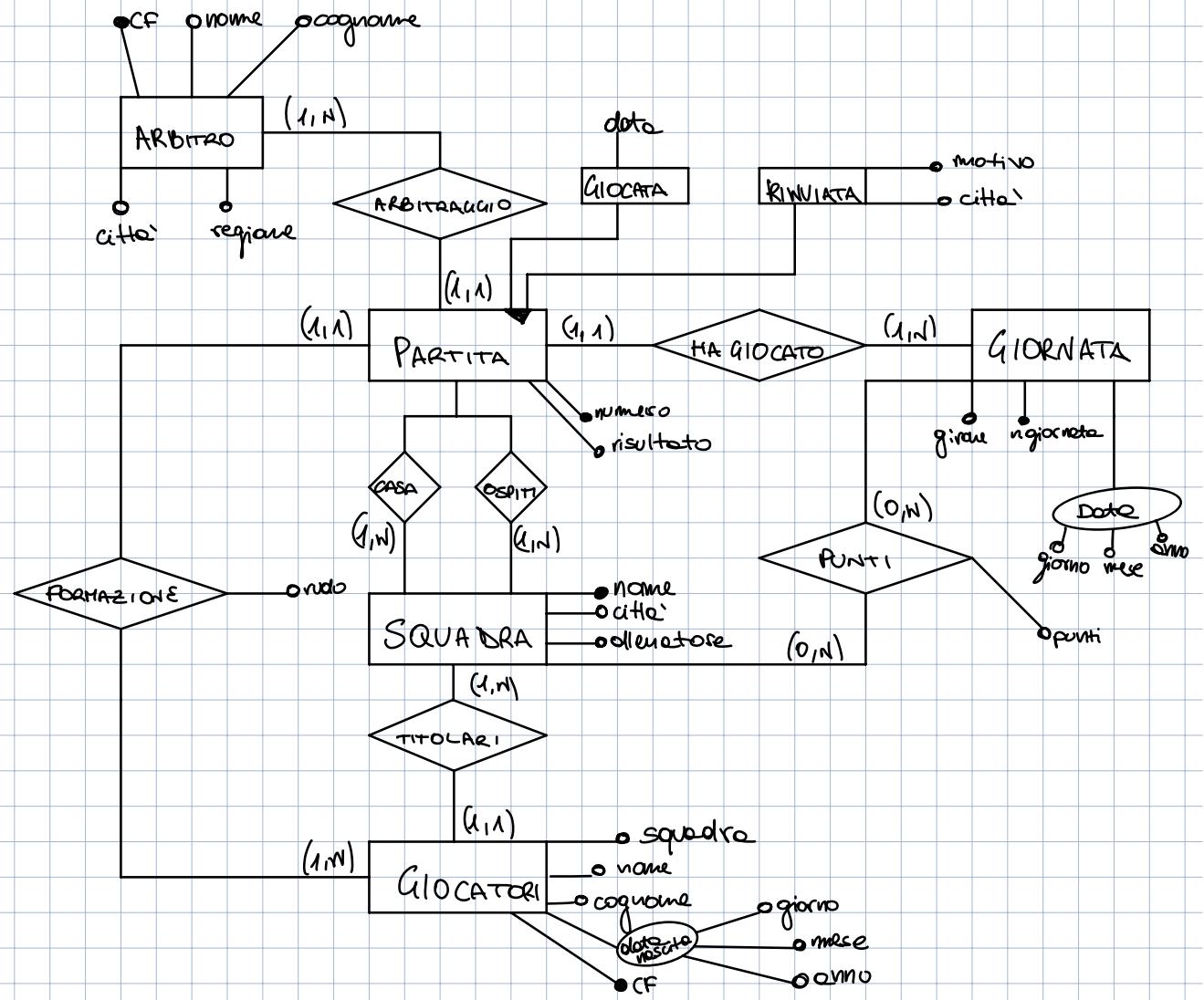
$\pi_{IdUtente, Nome, Cognome} (\sigma_{CATEGORIA = 1} (\chi_{IdUtente, Nome, Cognome})) \rightarrow CATEGORIA$
 $(Utente \bowtie Oroline))$

SE fosse STATO ALMENO 1 INVECE

$\pi_{IdUtente, Nome, Cognome} (\sigma_{CATEGORIA \geq 1} (\chi_{IdUtente, Nome, Cognome})) \rightarrow CATEGORIA$
 $(Utente \bowtie Oroline))$

Quesito 1





Posto (numero_giornata, risultato, giocato, Squadra_casa, Squadra_ospite, arbitro, rinvio, mese, data_rinvio)

Squadra (name, città, allenatore)

Giocatori (CF, nome, cognome, ruolo, GiornoW, MeseW, AnnoW, squadra)

Giornata (numero, giorno, mese, anno)

Arbitro (CF, nome, cognome, città, regione)

Punti (Squadra, giornata, punti)

Formazione (Giocatore, numero_giornata, giornata, ruolo)

Query secondo appello

a -

```
SELECT u.IdUtente, u.Email  
FROM Utente AS u JOIN Previsione AS p ON u.IdUtente = p.IdUtente  
WHERE p.Umidita = (SELECT min(Umidita)  
                     FROM Previsione)
```

b -

```
SELECT c.Nome, c.Regione  
FROM Citta AS c JOIN StazioneMeteo AS s ON c.Nome = s.Citta  
                  JOIN Previsione AS p ON s.IdStazione = p.IdStazione  
GROUP BY c.Nome, c.Regione  
HAVING avg(p.Temperatura) > (SELECT c.Stato, avg(p.Temperatura)  
                                 AS TempMedia  
                               FROM Citta AS c JOIN StazioneMeteo AS s  
                               ON c.Nome = s.Citta JOIN Previsione  
                               AS p ON s.IdStazione = p.IdStazione  
                               GROUP BY c.Stato  
                               HAVING max(TempMedia));
```

c -

```
SELECT c.Regione, COUNT(DISTINCT s.IdStazione) AS NumStazioni,  
      min(p.Temperatura), max(p.Umidita)  
FROM Citta AS c JOIN StazioneMeteo AS s ON c.Nome = s.Citta  
                  JOIN Previsione AS p ON s.IdStazione = p.IdStazione  
GROUP BY c.Regione
```

d -

```
SELECT c.Provincia, avg(p.Umidita)  
FROM Citta AS c JOIN StazioneMeteo AS sm ON c.Nome = sm.Citta  
                  JOIN Previsione AS p ON sm.IdStazione = p.IdStazione  
GROUP BY c.Provincia  
HAVING COUNT(sm.IdStazione) > 1
```

Query terzo appello

a -

$$\begin{aligned} & \text{SELECT COUNT(DISTINCT o.IdUtente, nome, cognome)} \\ & \quad \text{FROM Utente AS u JOIN Ordine AS o ON u.IdUtente = o.IdUtente} \\ & \quad \text{JOIN Prodotto AS p ON o.IdProdotto = p.IdProdotto} \\ & \quad \text{WHERE o.Data > '2022-01-01'} \\ & \quad \text{GROUP BY o.IdUtente, nome, cognome} \end{aligned}$$

Count (IdOrdine) \rightarrow Count (IdUtente, nome, cognome)

Countaggio (Utente e Ordine)

b -

```
SELECT CATEGORIA, AVG(PREZZO), MIN(PREZZO), MAX(PREZZO)
FROM PRODOTTO
GROUP BY CATEGORIA
```

c -

```
SELECT u.IdUtente, u.Email
FROM Utente AS u JOIN Ordine AS o ON u.IdUtente = o.IdUtente
JOIN Prodotto AS p ON o.IdProdotto = p.IdProdotto
WHERE o.Data > '2022-01-01'
GROUP BY u.IdUtente, u.Email
HAVING COUNT(o.IdProdotto) > 100
```

d -

```
SELECT p.IdProdotto, p.Nome, p.Prezzo
FROM Prodotto AS p JOIN Ordine AS o ON p.IdProdotto = o.IdProdotto
JOIN Utente AS u ON o.IdUtente = u.IdUtente
WHERE u.Nominativo = 'Genova'
GROUP BY p.IdProdotto, p.Nome, p.Prezzo
HAVING COUNT(DISTINCT o.IdOrdine) >= 1
```

```
SELECT p.IdProdotto, p.Nome, p.Prezzo
FROM Prodotto AS p JOIN Ordine AS o ON p.IdProdotto = o.IdProdotto
JOIN Utente AS u ON o.IdUtente = u.IdUtente
WHERE u.Nominativo = 'Genova'
GROUP BY p.IdProdotto, p.Nome, p.Prezzo
HAVING COUNT(DISTINCT o.IdOrdine) >= 1
```