

Si consideri la seguente base di dati per la gestione di stabilimenti balneari italiani di una catena:

Stabilimento(IdStabilimento, Indirizzo, Citta)

Prenotazione(IdStabilimento, CF-Cliente, Data, Costo)

A. Nel riquadro, scrivere una query in Algebra Relazionale che restituisca i codici fiscali dei clienti che nel 2023 hanno prenotato in esattamente un unico stabilimento della catena (2 punti).²

$$P1 = \sigma_{DATA \geq "01/01/2023" \text{ AND } DATA \leq "31/12/2023"}(P2 \text{ PRENOTAZIONE})$$

$$P2 = P1$$

$$P3 = \pi_{CF-CALIENTE}$$

$$[P1 \bowtie_{P1.IDSTABILIMENTO = P2.IDSTABILIMENTO} P2]$$

$$\text{AND } P1.CF-CALIENTE \neq P2.CF-CALIENTE$$

$$- \text{SARÀ PER ID STAB DIVERSO } [P1.IDSTAB <> P2.IDSTAB]$$

$$\pi_{CF-CALIENTE} P1 - P3_{CF-CALIENTE}$$

2.

Nel riquadro, scrivere una query in Standard SQL che restituisce l'identificativo dello stabilimento che, in media, giornalmente ha più clienti.

```
CREATE VIEW Clienti_per_giorno AS
SELECT IdStabilimento, Data, COUNT(*) AS Num
FROM Prenotazione
GROUP BY IdStabilimento, Data;
```

```
CREATE VIEW Media_giornaliera_clienti AS
SELECT IdStabilimento, Data
FROM Clienti_per_giorno
WHERE Num = (SELECT AVG(Num) FROM Clienti_per_giorno);
```

```
SELECT IDStabilimento, Data
FROM Media_giornaliera_clienti
WHERE Media_Clienti = (SELECT MAX(Media_Clienti
                           FROM Media_giornaliera_clienti));
```

// Alternativamente

```
CREATE VIEW Media_giornaliera_clienti AS
SELECT IdStabilimento, Data, AVG(Num) AS Media_Clienti
FROM Prenotazione
GROUP BY IdStabilimento, Data;
```

3. Nel riquadro, scrivere una query in Standard SQL che restituisce, per ogni stabilimento con identificativo S a Rimini, una coppia (S,N) dove N è il numero di clienti diversi che hanno prenotato nello stabilimento con id S

```
CREATE VIEW Clienti_per_stabilimento (N_clienti, IdStab) AS
SELECT COUNT(CF-Cliente), IdStabilimento
FROM Prenotazione
GROUP BY IdStabilimento;
```

```
SELECT DISTINCT(N_clienti), IdStab
FROM Clienti_per_stabilimento CS, Prenotazione P
WHERE CS.IdStab = P.IdStabilimento
AND Citta = "Rimini"
GROUP BY IdStab;
```



1. RIPULSA OPERAZIONI

DA FINIS A CK (TS, TG)

$$\text{RSDO} = \{T8\} \rightarrow C(T8)$$

$$\text{UNDO} = \{TS, TG\}$$

\uparrow

$$\{TS, TG, T7\} \rightarrow U(T7 \dots)$$

$$\text{RSDO} = \{T8\}$$

$$\text{UNDO} = \{TS, TG, T7\}$$

~~T7?~~

\cup

$$\left. \begin{array}{l} TG? \rightarrow O_3 = B_7 \\ T7? \rightarrow O_6 = B_6 \end{array} \right\} \text{UNDO}$$

$$T8? \rightarrow [O_5 = A_5] \cup \dots$$

RSDO

$R(\underline{A}, \underline{B}, C, D)$

DF : $A \rightarrow BD$,

$B \rightarrow A$

$B \rightarrow C$

$A^+ = \{A, B, C, D\}$

$B^+ = \{B, A, C, D\}$

$R_1 \rightarrow (\underline{A}, B, C, D)$

$R_2 \rightarrow (\underline{B}, A, C, D)$

$X^+ \subset Y^+ \quad \nrightarrow$

$R_1(\underline{A}, \underline{B}, C, D)$

$R_2(\underline{A}, B, D)$
 $R_3(\underline{B}, A, C)$

BCNF

$\left\{ \begin{array}{l} R_1(\underline{A}, B, D) \\ R_2(\underline{B}, A) \\ R_3(\underline{B}, C) \end{array} \right\}$

3FN

$X \not\rightarrow$

SUPERFLUOUS



$\underline{A} \quad B$ QUERY
 $\leq \textcircled{2} \rightarrow K = \text{DISTINCT} \dots$

$$\underline{A} \circ B \subseteq C \circ D$$

$$\leq R$$

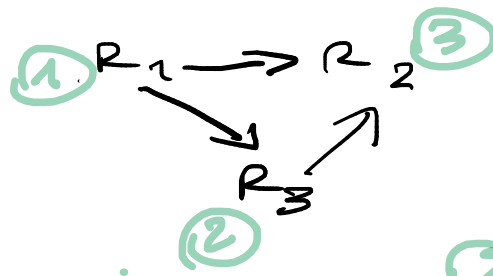
$$= R \rightarrow A \circ B$$

$\emptyset \rightarrow \text{otherwise}$

Si consideri lo schedule $S = \{r_1(a) \ w_2(a) \ r_3(a) \ w_1(b) \ w_3(c) \ r_2(d) \ r_3(d) \ w_3(b) \ w_3(e)\}$

Dopo aver calcolato il grafo dei conflitti, indicare se S è conflict-serializzabile (CSR) e/o view-serializzabile (VSR)?

1. R è in CSR **ma non** in VSR
2. R **non** è in CSR **ma** è in VSR
3. R è in CSR ed è in VSR
4. R **non** è in CSR e non è VSR



NO CICLO = CSR \subseteq VSR

CONFLICT VIEW-EQUIVALENCE

- ①
 - $R_1(a)$
 - $w_1(b)$
- ②
 - $R_3(a)$
 - $w_3(c)$
 - $R_3(d)$
 - $w_3(b)$
- ③
 - $w_2(a)$
 - $R_2(d)$

Esercizio 4: Normalizzazione (5 punti)

Dato lo schedule $S = r_1(x) \ r_1(t) \ r_2(z) \ w_3(x) \ w_1(x) \ r_1(y) \ w_3(t) \ w_2(x) \ w_1(y)$, indicare se S è view e/o conflict-serializzabile, motivando la risposta. Se S è view e/o conflict-serializzabile, indicare uno schedule seriale che è view e/o conflict-equivalente, motivando la risposta.



NO CSR \rightarrow CICLO

$R_1(x) \ R_1(t) \ w_1(x) \ R_1(y) \ w_1(y) \rightarrow R_1$

$w_3(x) \ w_3(t) \rightarrow R_3 \ / \ R_2(z) \ w_2(x) \rightarrow R_2$

Esercizio 4: Normalizzazione (6 punti)

Sia data la relazione $R(A,B,C,D,E,F,G,H,I)$ con copertura ridotta $G=\{AB \rightarrow C, AD \rightarrow GH, BD \rightarrow EF, A \rightarrow I\}$

- Trovare la/e chiave/i candidata/e di R , motivando la risposta.
- Quali dipendenze violano la 3NF? Motivare la risposta.
- Effettuare una decomposizione in 3NF.

$$\left\{ \begin{array}{l} AB^+ = \{A, B, C, I\} \\ AD^+ = \{A, D, G, H\} \\ BD^+ = \{B, D, E, F\} \\ A^+ = \{A, I\} \end{array} \right.$$

$$AB^+ \cup AD^+ \cup BD^+ \\ \rightarrow \underline{ABD} \\ = \\ \text{CHIAVE CANDIDATA}$$

② \Rightarrow 3FN?

$\rightarrow X \subseteq Y$ con X NON PRIMA CAUSA
 X, D, F NON PRIMA CAUSA

DECOMPOSIZIONE

1. SUBSETS $X \rightarrow A, Y \rightarrow B \cup \dots$
 $X^+ = Y^+$

$\{AB \rightarrow C\}, \{AD \rightarrow GH\}, \{BD \rightarrow EF\},$
 $\{A \rightarrow I\}$

2. RELAZIONE $\forall \sum_{i=1}^n S_i$



$R_1(A, B, C)$ / $R_2(A, D, G, H)$, $R_3 = d(B, D, E, F)$
 $R_4(A, E)$

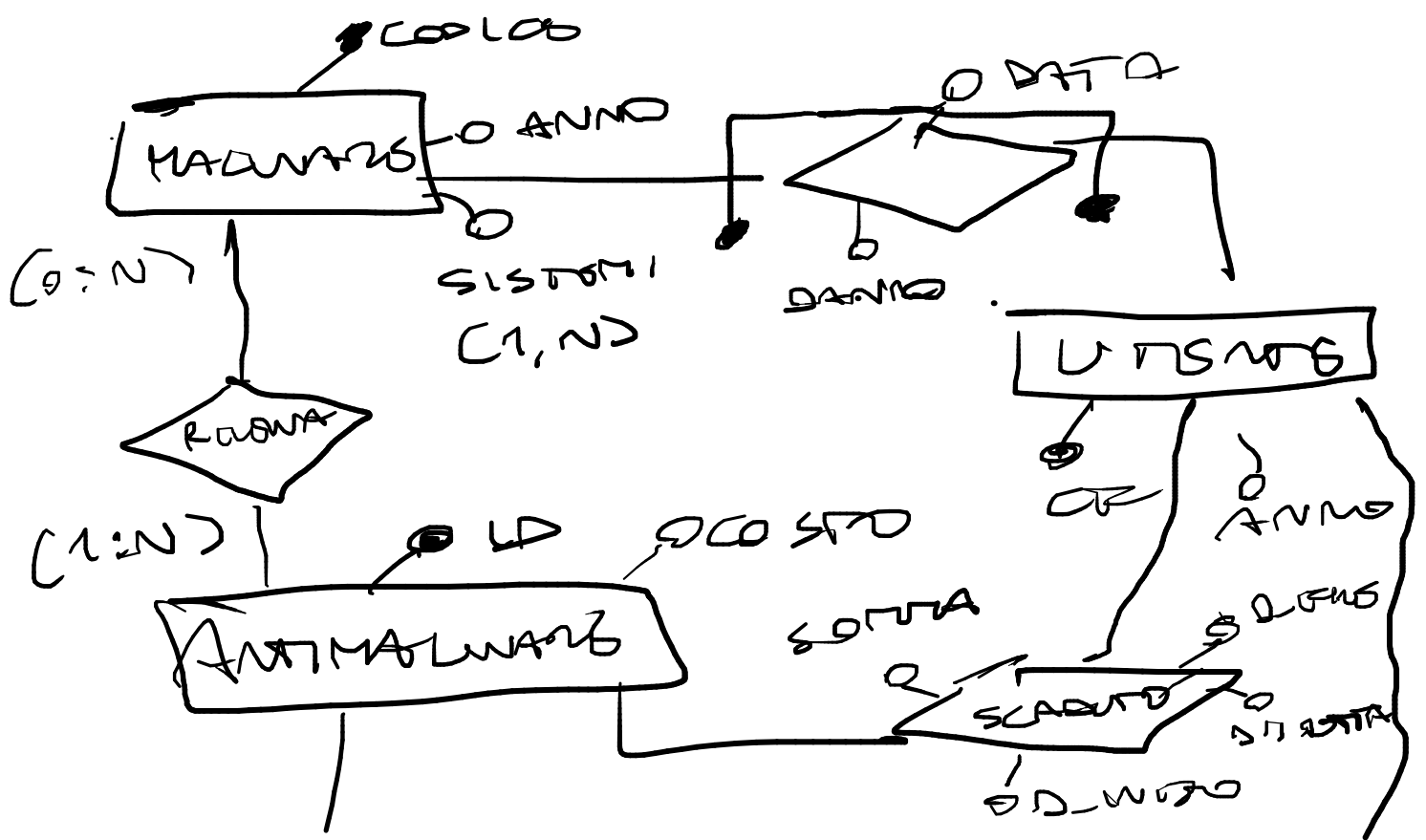
3. $A(x) \in T(y)$, $x \leq y$, ~~A~~
NO

4. CHIAUS SONZA RELAZIONE $\rightarrow R_5(A, B, D)$

$R_1(A, B, C)$ / $R_2(A, D, G, H)$ / $R_3 = d(B, D, E, F)$
 $R_4(A, E)$ / $R_5(A, B, D)$

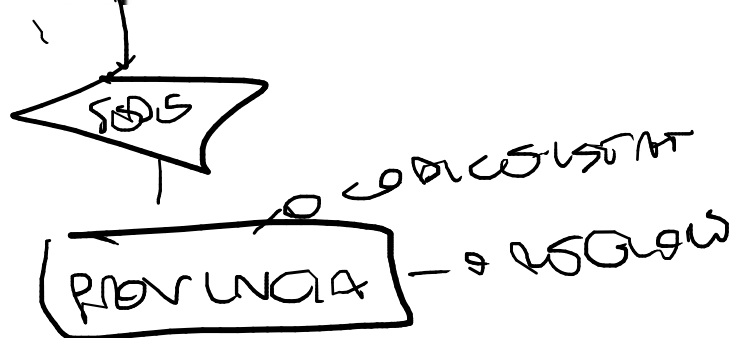
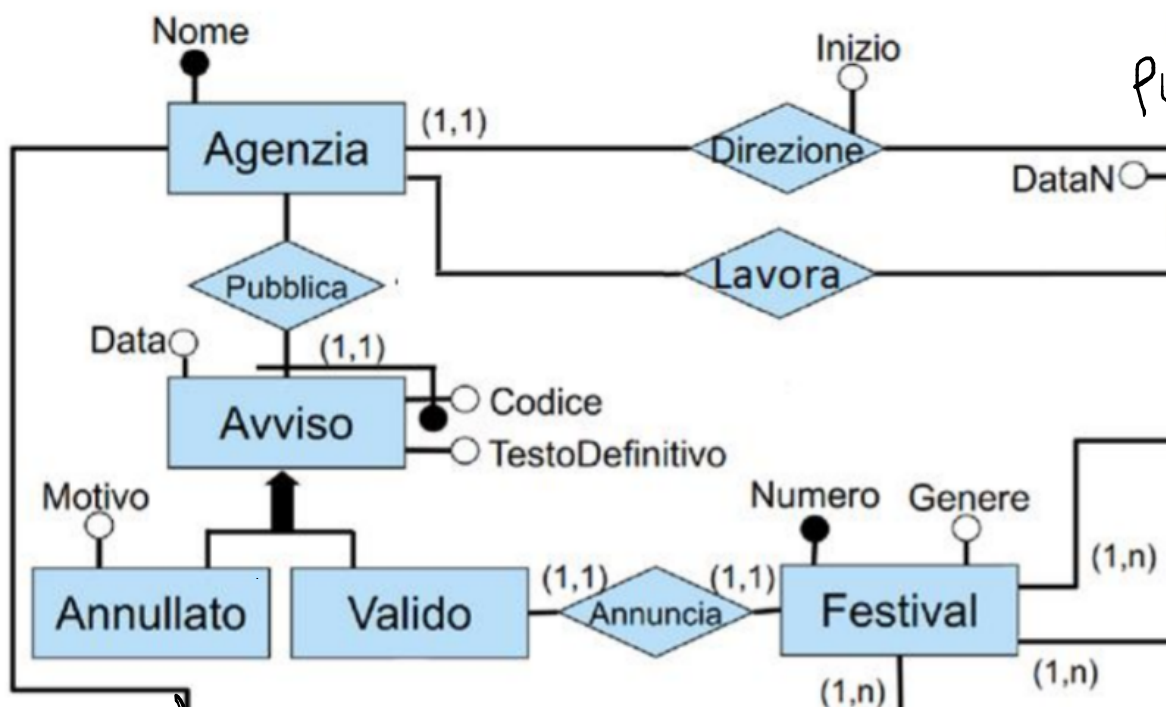
BCNF ? $\rightarrow \textcircled{Y} \rightarrow$ BEN \in BCNF (V)

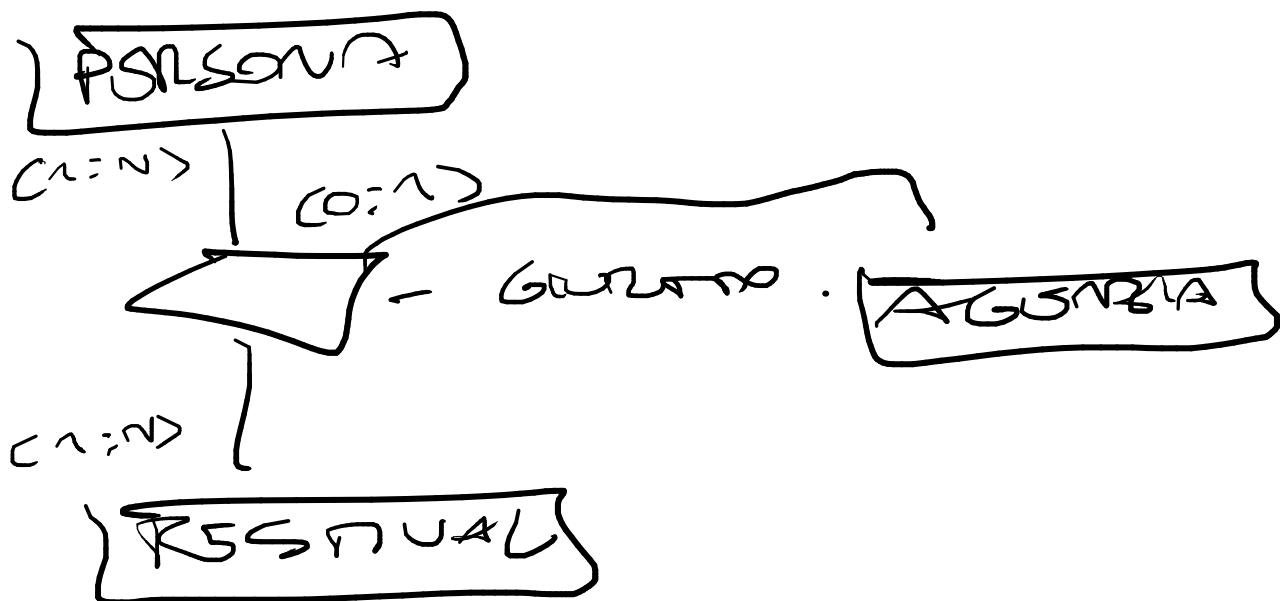
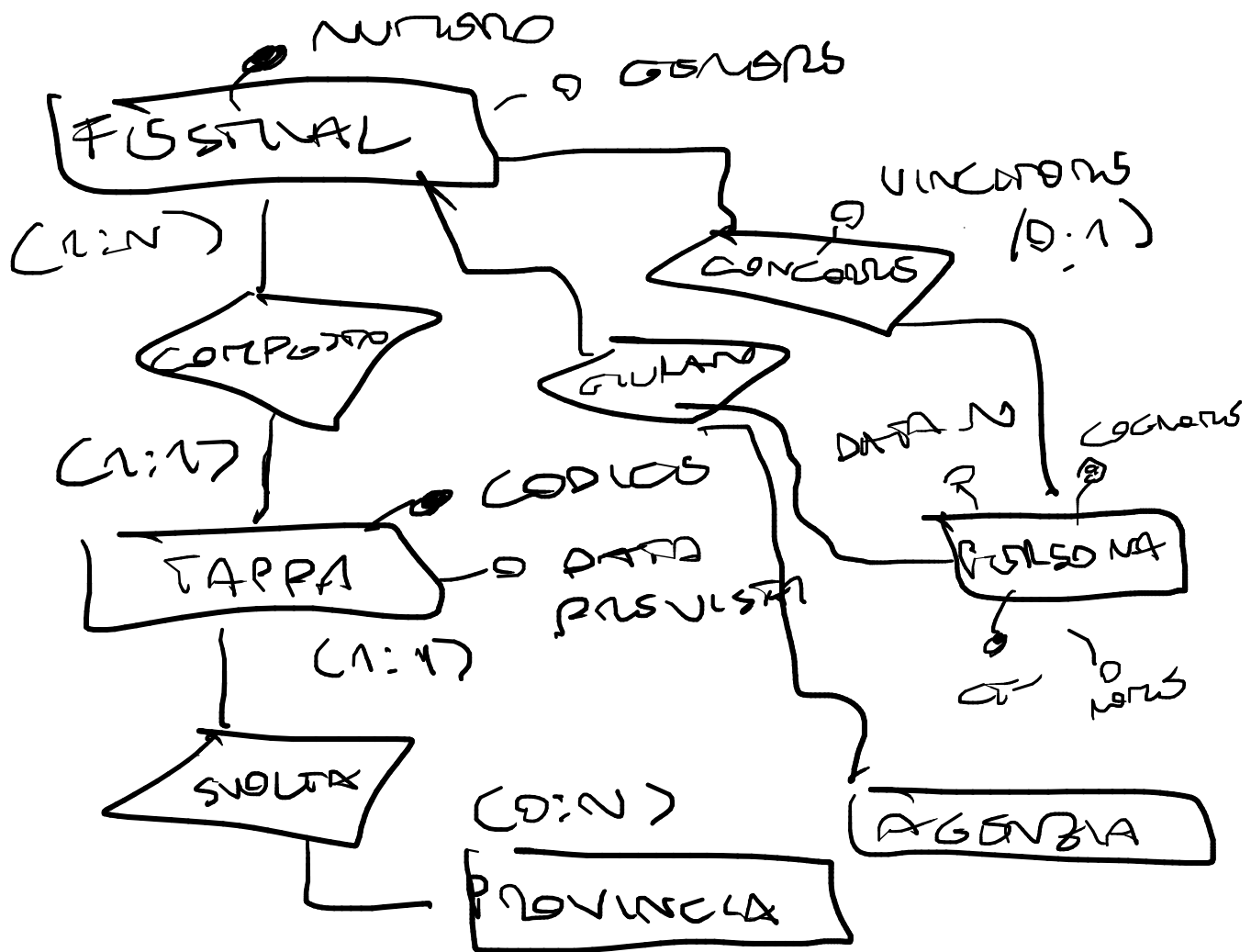
$\{A \rightarrow C\}$
 $\{AD \rightarrow GH\}$
 $\{BD \rightarrow EF\}$
 $\{A \rightarrow E\}$

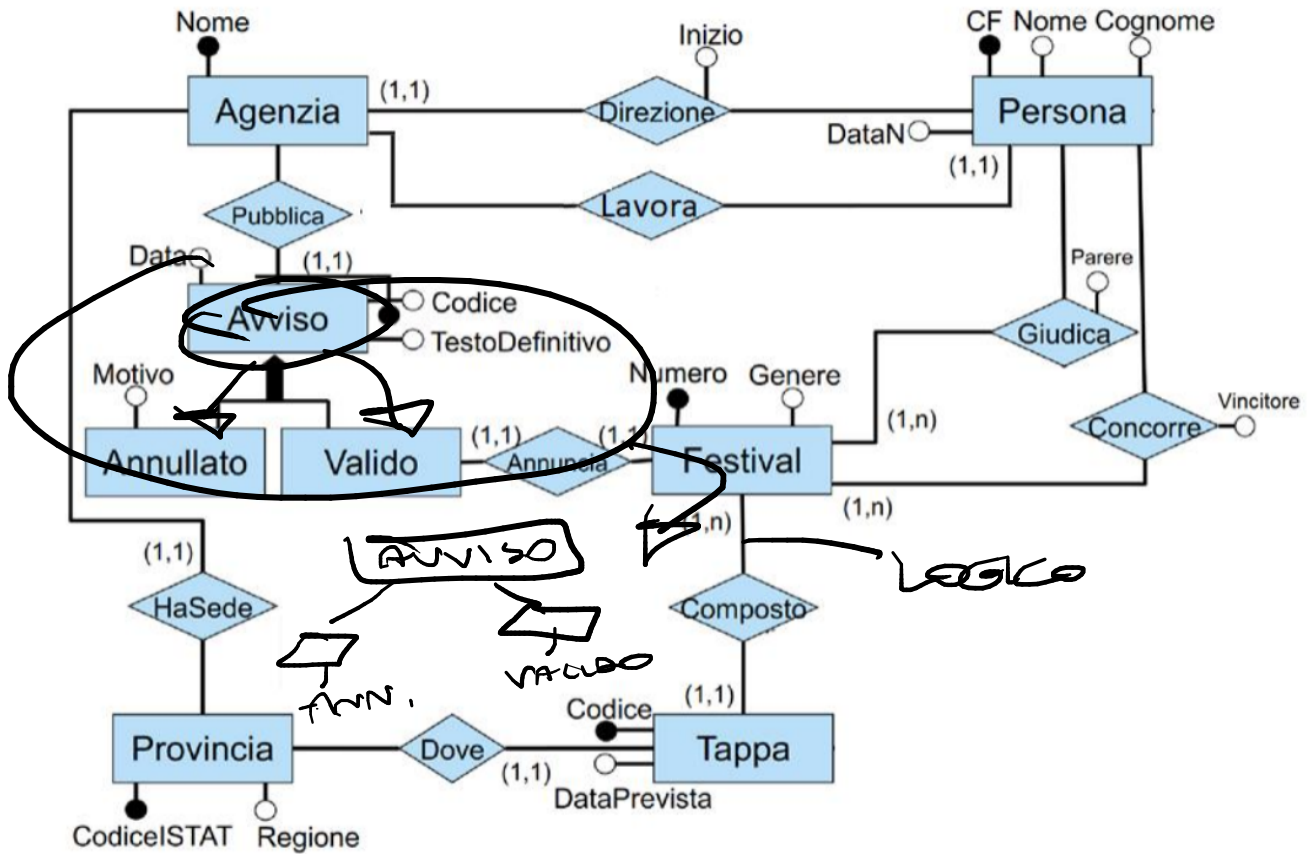


DATA - IN 7th









IDEAS ---

$\Pi_{CF} (LAVORA / IMPUGNATO)$

→ STRESSO CF

AND DATAFUS IS NULL

↓

LAVORA ANCORA

$\Pi_{CF} LAVORA \bowtie$

LAVORA.CF = IMPUGNATO.CF AND
DATAFUS IS NULL

IMPUGNATO

$$L_1 =$$

π_{CF} LAJORA ~~X~~ IMPUSGADO
 LAJORA.CF = AND DATAFMS
 IMPUSGADO.CF IS NULL

$$L_2 = L_1$$

$\pi_{L_1.CF}(L_1) =$

$\pi_{L_1.CF}(L_1 \times L_2)$

$L_1.PIVA \subset L_2.PIVA$

AND

$$L_1.CF = L_2.CF$$

Nel riquadro, scrivere una query in Standard SQL che,
C. per ogni partita IVA in Veneto relativo ad aziende
con almeno 10 dipendenti,
restituisce lo stipendio medio

```
CREATE VIEW Stipendio_per_azienza (Azienda, Medio) AS  
SELECT PIVA, AVG(StipendioMensile)  
FROM Lavora AS L, Azienda AS A  
WHERE Regione = "Veneto"  
AND L.PIVA = A.PIVA  
GROUP BY PIVA  
HAVING COUNT(CF) ≥ 10;
```

// O'Lions

```
SELECT PIVA, AVG(STIPENDIOMENSILE)  
FROM LAVORA  
WHERE DATAFINE IS NULL AND PIVA IN  
(SELECT PIVA FROM AZIENDA WHERE REGIONE='Veneto')  
GROUP BY PIVA  
HAVING COUNT(CF)>9
```

B. Scrivere una query in Standard SQL
che restituisce la partita IVA dell'azienda
con più dipendenti impiegati alla data attuale

```
CREATE VIEW Numero_dipendenti (Numero, Azienda) AS  
SELECT COUNT(CF), PIVA  
FROM Lavora  
WHERE Datafine IS NULL  
GROUP BY PIVA;
```

```
SELECT Azienda  
FROM Numero_dipendenti  
WHERE Numero = (SELECT MAX(Numero)  
                FROM Numero_dipendenti);
```

Restituire il nome del candidato che si sono presentati una sola volta (cioè in un solo anno) per il PD (2 punti).

```
C1 =  $\sigma_{\{Partito = "PD"\}}$  CANDIDATURA  
C2 = C1
```

```
DUE_CANDIDATURE =  
 $\pi_{\{C1.Candidato\}}$   
("C1"  $\bowtie_{\{C1.Candidato = C2.Candidato \text{ AND } C1.Partito \neq C2.Partito \text{ AND } C1.Anno = C2.Anno\}}$  "C2")
```

```
 $\pi_{\{C1.Candidato\}}(C1) \setminus DUE\_CANDIDATURE$ 
```

B. Scrivere una query in Standard SQL che restituisce, per ogni partito che ha ottenuto almeno il 10% a Padova nel 2022, il numero medio di voti tra tutti i candidati del 2022

```
SELECT Partito, AVG(NumVoti)  
FROM Candidatura C, Risultato-Partito RP  
WHERE C.Partito = RP.Partito  
AND C.Anno = RP.Anno  
AND Percentuale  $\geq$  10  
AND Città = "Padova"  
AND Anno = 2022  
GROUP BY Partito;
```

```
// O'Lions
```

```
SELECT PARTITO, AVG(NUMVOTI)  
FROM CANDIDATURA C  
WHERE ANNO=2022 AND CITTA='Padova'  
AND PARTITO IN  
(SELECT PARTITO FROM RISULTATO-PARTITO  
WHERE PERCENTUALE $\geq$ 10 AND ANNO=2022  
AND CITTA='Padova')  
GROUP BY PARTITO
```

C. Scrivere una query in Standard SQL che, per ogni città con votazioni nel 2022, restituisce il partito che ha ottenuto più voti

```
CREATE VIEW Voti_partito AS  
SELECT Partito, Città, SUM(NumVoti)  
FROM Risultato-Partito RP  
JOIN Candidatura C ON RP.Anno = C.Anno  
WHERE Anno = 2022  
GROUP BY Partito, Città
```

Il nome dei fantini che hanno vinto
solamente gare con cavalli della
stessa scuderia

$$J_1 = \text{GARE} \bowtie \text{CAVALLI} \\ \text{CAVALLOVINCENTE} \\ = \\ \text{CAVALLI.NOME}$$

$$J_2 = J_1$$

$$\pi_{\text{FANTINOVINCENTE}} J_1 \bowtie \\ J_2.FANTINOVINCENTE \\ = \\ \text{FANTINO} \\ \text{FANTINO.NOME}$$

$$\bowtie \left[\begin{array}{l} J_1.FANTINOVINCENTE \\ = \\ J_2.FANTINOVINCENTE \end{array} \right] J_2 \\ \text{AND}$$

$$J_1.SCUDERIA = \\ J_2.J_2.SCUDERIA$$

$$J1 = \text{Gare} \bowtie \text{CavalloVincente} = \text{Cavalli.Nome} \text{ Cavalli}$$

$$J2 = J1$$

$$\pi_{\text{FantinoVincente}} (J1) -$$

$$\pi_{J1.FantinoVincente} (J1 \bowtie$$

$$J1.Scuderia <> J2.Scuderia \text{ AND } J1.FantinoVincente = J2.FantinoVincente \text{ } J2)$$