

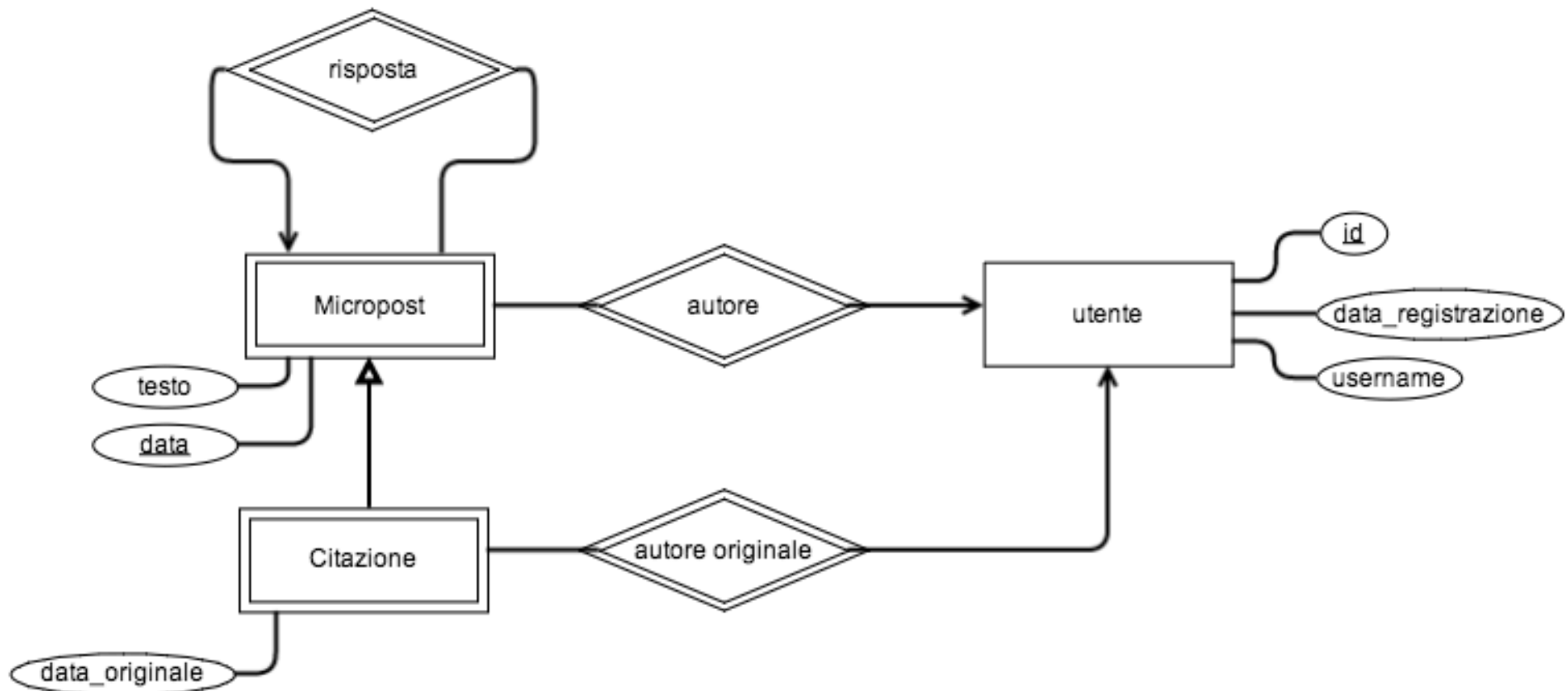
# Suggerimenti per le soluzioni dell'Esame

*21/01/2014*

Corso di Basi di Dati  
A.A. 2013/2014

Matteo Lissandrini  
ml@disi.unitn.it

## Esercizio 1 – Soluzioni (1/3)



# Esercizio 1 – Soluzioni (2/3)

Utente (id, username, data)  
Micropost (data, autore, testo)  
Risposta (data\_risposta, autore\_risposta, data\_soggetto, autore\_soggetto)  
Citazione (data\_citazione, autore\_citazione, data\_originale, autore\_originale)

```
CREATE TABLE exam."Micropost"  
(  
    data TIMESTAMP,  
    autore INT REFERENCES exam."Utente"(id),  
    testo VARCHAR(160),  
  
    PRIMARY KEY (data, autore)  
);
```

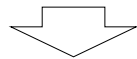
# Esercizio 1 – Soluzioni (3/3)

```
INSERT INTO exam."Micropost"  
VALUES (CURRENT_TIMESTAMP, 1, 'This is a micropost by UserID 1' );
```

*si deve assumere che l'utente esista*

## Esercizio 2 – Soluzioni (1/3)

$\pi_{\text{nome, anno, lingua}}$  (  
 $\sigma_{\text{mese}=1 \text{ AND } ((\text{anno} > 2002 \text{ AND anno } 2007) \text{ OR anno} = 2010)}$  ( Utente )  
)



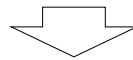
```
SELECT nome, anno_reg, lingua
FROM Utente
WHERE mese_reg = 1
      AND( ( anno_reg > 2002 AND anno_reg < 2007 )
          OR anno_reg = 2010 )
```

## Esercizio 2 – Soluzioni (2/3)

```

τU1.nome ( δ ( πU1.nome (
    (
        ρU1 (nome, anno, mese) ( πnome, anno_reg, mese_reg ( Utente ) )
        ⋈U1.nome = Segue.nome_segue
        ( Segue )
    )
    ⋈Segue.nome_seguito = U2.nome AND ( U1.anno < U2.anno OR ( U1.anno = U2.anno AND U1.mese < U2.mese ) )
    ( ρU2 (nome, anno, mese) ( πnome, anno_reg, mese_reg ( Utente ) ) )
)))

```



```

SELECT DISTINCT u1.nome
FROM "Utente" AS u1
    JOIN "Segue" ON u1.nome = "Segue".nome_segue
    JOIN "Utente" AS u2 ON u2.nome = "Segue".nome_seguito
WHERE u1.anno_reg < u2.anno_reg
    OR ( u1.anno_reg = u2.anno_reg AND u1.mese_reg < u2.mese_reg )
ORDER BY u1.nome DESC

```

## Esercizio 2 – Soluzioni (3/3)

$$\begin{aligned}
 & \gamma_{U2.anno, AVG(numero) \rightarrow numero} ( \\
 & \quad \gamma_{U2.nome, COUNT(Segue.nome_seguito) \rightarrow numero} ( \\
 & \quad \quad ( \\
 & \quad \quad \quad \rho_{U1(nome)} (\pi_{nome} (\sigma_{anno\_reg > 2009 \text{ AND } anno\_reg < 2013} (\pi_{nome, anno\_reg} (Utente)))) - v1) \\
 & \quad \quad ) \bowtie_{U1.nome = Segue.nome\_segue} (Segue) \\
 & \quad ) \bowtie (\rho_{U2(nome, anno)} (\pi_{nome, anno\_reg} (Utente))) \\
 & )
 \end{aligned}$$

```

SELECT u.anno_reg AS anno, AVG(seguiti.numero)
FROM ( SELECT uss.nome, COUNT(s.nome_seguito) AS numero
      FROM (
        SELECT u1.nome FROM Utente AS u1
        WHERE u1.anno_reg > 2009 AND u1.anno_reg < 2013
        EXCEPT SELECT v1.nome FROM v1
      ) AS uss JOIN Segue AS s ON uss.nome = s.nome_segue
    GROUP BY uss.nome ) AS seguiti
JOIN Utente AS u ON seguiti.nome = u.nome
GROUP BY u.anno_reg
ORDER BY u.anno_reg

```

## Esercizio 3 – Soluzioni (1/2)

```
DELETE FROM Utente AS u
WHERE u.nome IN (
    (
        ( SELECT u1.nome FROM Utente AS u1
          WHERE u1.anno_reg < 2005
        ) EXCEPT SELECT DISTINCT s.nome_segue
              FROM Segue AS s
        )
    EXCEPT SELECT DISTINCT s.nome_seguito
              FROM Segue AS s
)
```



## Esercizio 3 – Soluzioni (2/2)

```
WITH RECURSIVE mass AS (  
    SELECT ss.nome_segue AS nome  
    FROM Segue AS ss  
    JOIN (  
        SELECT s.nome_seguito AS nome  
        FROM Segue AS s  
        GROUP BY s.nome_seguito  
        HAVING COUNT(s.nome_segue) = (  
            SELECT MAX(seguiti.numero)  
            FROM (  
                SELECT s.nome_seguito ,  
                       COUNT(s.nome_segue) as numero  
                FROM Segue AS s  
                GROUP BY s.nome_seguito ) AS seguiti  
            )  
        ) AS ff  
    ON ss.nome_seguito = ff.nome  
    UNION ALL  
    (SELECT s1.nome_segue AS nome  
     FROM mass JOIN Segue AS s1  
     ON mass.nome = s1.nome_seguito  
    )  
)  
SELECT mass.nome FROM mass
```

## Esercizio 4 – Soluzioni (2/2)

```
SELECT A, MIN(B) FROM R GROUP BY A
```

```
NULL ; 2  
4     ; NULL  
7     ; 2
```

```
SELECT COUNT(*), COUNT(A) FROM R
```

```
5     ; 4
```

```
SELECT A, B FROM R WHERE A = C
```

```
4     ; NULL
```

```
SELECT B, C FROM R WHERE A <> C
```

```
NULL ; 8
```

## Esercizio 5 – Soluzioni (1/2)

	{R}	{S}	{W}	{U}	R(a,b)	S(b,c)	W(c,d)	U(d,e)
Dimensione	100	200	200	300				
Costo	0	0	0	0				
Migliore	{R}	{S}	{W}	{U}				
	{R,S}	{R,W}	{R,U}	{S,W}	{S,U}	{W,U}		
Dimensione	2000	20 000	30 000	8000	60 000	600		
Costo	0	0	0	0	0	0		
Migliore	(R ⋈ S)	(R ⋈ W)	(R ⋈ U)	(S ⋈ W)	(S ⋈ U)	(W ⋈ U)		
	{R,S,W}	{R,S,U}	{R,W,U}	{S,W,U}				
Dimensione	80000	600 000	60 000	24 000				
Costo	2000	2000	20 000	600				
Migliore	(R ⋈ S) ⋈ W	(R ⋈ S) ⋈ U	(R ⋈ W) ⋈ U	S ⋈ (W ⋈ U)				

$R \bowtie (S \bowtie (W \bowtie U)) = 24\ 000 + 600$   
 $(R \bowtie S) \bowtie (W \bowtie U) = 2000 + 600$   
 $T(R \bowtie S) = T(R) * T(S) / V(R,b) = 100 * 200 / 10 = 2000$   
 $T(W \bowtie U) = T(W) * T(U) / V(U,d) = 200 * 300 / 100 = 600$

## Esercizio 5 – Soluzioni (2/2)

$$\mathbf{a) \delta( \sigma_{W.d = U.d} ( \pi_{W.d}(W) \times \pi_{U.d}(\sigma_{U.e = 11}(U)) ) )}$$

$$T(\pi_{W.d}(W)) = 200$$

$$T(\pi_{U.d}(\sigma_{U.e = 11}(U))) = 300/50 = 6$$

$$T(\pi_{W.d}(W) \times \pi_{U.d}(\sigma_{U.e = 11}(U))) = 200 * 6 = 1200$$

$$\mathbf{b) \delta( \pi_{W.d}(W) ) \bowtie_{W.d = U.d} \delta( \pi_{W.d} ( \sigma_{U.e = 11}(U) ) )}$$

$$T(\pi_{W.d}(W)) = 200$$

$$T(\delta(\pi_{W.d}(W))) = 5$$

$$T(\delta(\pi_{U.d}(\sigma_{U.e = 11}(U)))) = 300/50 = 6$$

$$T(\delta(\pi_{W.d}(W)) \bowtie_{W.d = U.d} \delta(\pi_{W.d} ( \sigma_{U.e = 11}(U) ) ) ) = 5 * 6 / 6 = 5$$

## Esercizio 6 – Soluzioni (1/2)

**R -<sub>B</sub> S**     $B(R) < M$

Leggi il contenuto di R in  $m_1 \dots m_M$   
(costruendo una appropriata struttura di ricerca in memoria)

Leggi un blocco da S

Per ogni tupla  $s$  in S, se  $s$  si trova in uno dei blocchi  $m_i$  allora rimuovi la tupla da  $m_i$

Ripeti il precedente fino a che tutto S non è stato letto

Scrivi in output ciò che rimane in  $m_1 \dots m_M$

## Esercizio 6 – Soluzioni (2/2)

**R -<sub>B</sub> S**     $B(R) > M$

Deve essere  **$B(R) + B(S) < M^2$**

Si ordinano entrambe R e S rispetto ad ogni attributo

Si legge il primo blocco da R,  $R_b$ , e da S,  $S_b$

Presa la prima tupla  $r$  da  $R_b$  si rimuovono tutte le tuple  $s$  da  $S_b$  tali per cui  $r < s$   
Se si raggiunge  $r = s$  si rimuove  $r$  da  $R_b$  e  $s$  da  $S_b$  e si ripete.

Se si raggiunge  $r > s$  si scrive in output  $r$  da  $R_b$ , si rimuove  $s$  da  $S_b$  e si ripete.

In qualsiasi momento se  $R_b$  o  $S_b$  risulta vuoto si legge il prossimo blocco.

Si continua per ogni blocco di R

## Esercizio 7 – Soluzioni (1/2)

a)

$T2 \rightarrow T1 : r2(C) \rightarrow w1(C)$

$T1 \rightarrow T2 : r1(A) \rightarrow w2(A)$

b)

$T1 \rightarrow T2 : r1(A) \rightarrow w2(A)$

## Esercizio 7 – Soluzioni (2/2)

- 1)  $ul_1(B) \ r_1(B) \ xl_1(B) \ w_1(B) \ w_1(B) \ u_1(B)$
- 2)  $xl_2(B) \ w_2(B) \ sl_2(A) \ r_2(A) \ u_2(B) \ u_2(A)$
- 3)  $ul_3(A) \ r_3(A) \ xl_3(A) \ w_3(A) \ u_3(A)$

T1	T2	T3
$ul(B); r(B)$		
		$ul(A); r(A)$
$xl(B); w(B); w(B)$		
	$xl(B); Denied$	
$u(B);$		
	$xl(B); w(B)$	
	$sl(A); Denied$	
		$xl(A); w(A); u(A)$



## Esercizio 8 – Soluzioni (1/1)

$A \rightarrow B, CD \rightarrow A, BC \rightarrow D, AE \rightarrow F, CD \rightarrow E$

**Deriva**

$CD \rightarrow B$

**Chiusure Transitive**

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

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

$\{AE\}^+ = \{A, E, F, B\}$

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

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

**Chiavi per R:**

AC o BC o CD

**BCNF:**

$R_1\{A, B\}$

$R_2\{A, E, F\}$

$R_3\{A, C, D, E\}$