

**Tempo a disposizione: 2:30 ore**

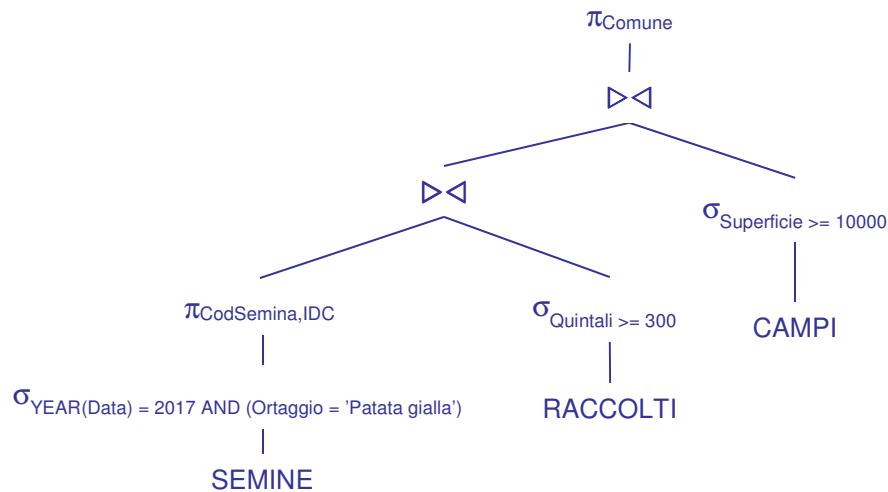
**1) Algebra relazionale (3 punti totali):**

Date le seguenti relazioni:

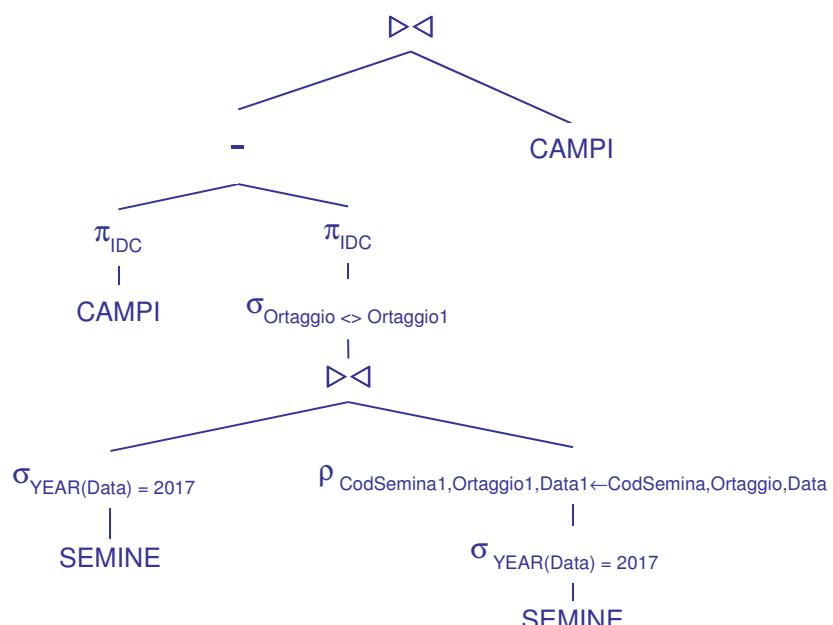
CAMP I (IDC, Superficie, Comune);  
 SEMINE (CodSemina, IDC, Ortaggio, Data),  
 IDC references CAMP I;  
 RACCOLTI (CodSemina, Data, Quintali),  
 CodSemina references SEMINE;  
 -- Superficie (espressa in metri quadri, mq) è di tipo INT  
 -- Quintali è di tipo DEC(6,2)

si scrivano in algebra relazionale le seguenti interrogazioni:

**1.1) [1 p.]** I comuni in cui nel 2017 è stata seminata la 'Patata gialla' in un campo di almeno 10000 mq e il cui raccolto ha prodotto almeno 300 quintali



**1.2) [2 p.]** I dati dei campi in cui nel 2017 è stato seminato un solo tipo di ortaggio



## Sistemi Informativi T

9 luglio 2018

Risoluzione

### 2) SQL (5 punti totali)

Con riferimento al DB dell'esercizio 1, si scrivano in SQL le seguenti interrogazioni:

- 2.1) [2 p.] Per ogni ortaggio, la quantità media ricavata per unità di superficie (ad es. se si hanno 2 raccolti, uno di 100 quintali in un campo da 1000 mq e l'altro di 240 quintali in un campo da 1200 mq, la media è:  $(100/1000 + 240/1200)/2 = (0.1 + 0.2)/2 = 0.15$  quintali/mq)

```
SELECT      S.Ortaggio, AVG(R.Quintali/C.Superficie) AS QtaMedia
FROM        CAMPI C, SEMINE S, RACCOLTI R
WHERE       C.IDC = S.IDC
AND         S.CodSemina = R.CodSemina
GROUP BY    S.Ortaggio;
```

- 2.2) [3 p.] Gli ortaggi che nel 2018 sono stati seminati almeno 2 volte in tutti i comuni

```
WITH
OCDUE (Ortaggio, Comune) AS (
SELECT      S.Ortaggio, C.Comune
FROM        CAMPI C, SEMINE S
WHERE       C.IDC = S.IDC
AND         YEAR(S.Data) = 2018
GROUP BY    S.Ortaggio, C.Comune
HAVING     COUNT(*) >= 2          )

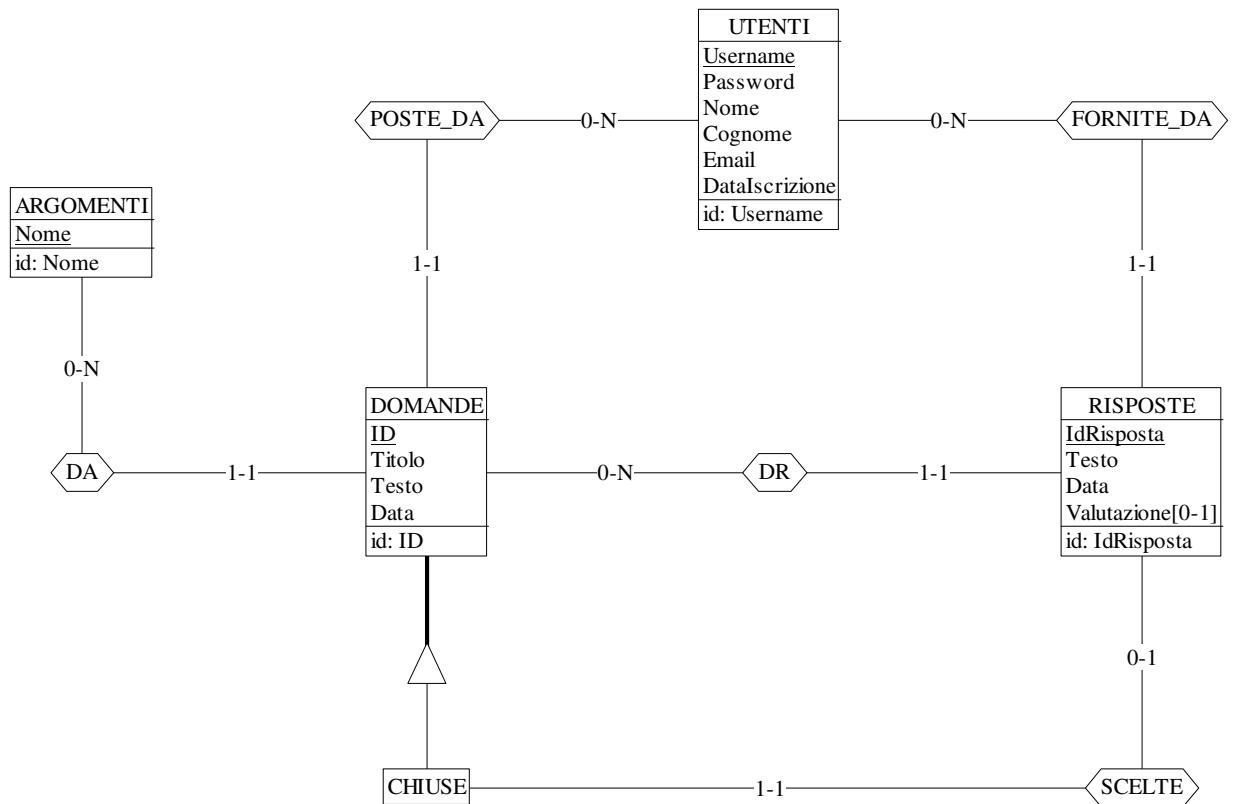
SELECT      O.Ortaggio
FROM        OCDUE O
GROUP BY    O.Ortaggio
HAVING     COUNT(*) = ( SELECT COUNT(DISTINCT C.Comune)
                        FROM    CAMPI C
                    );
```

-- La c.t.e. restituisce, per ogni ortaggio, i comuni in cui nel 2018  
-- l'ortaggio è stato seminato almeno 2 volte.  
-- Si noti l'uso di COUNT(DISTINCT C.Comune) per contare quanti sono  
-- i comuni presenti nel DB

### 3) Progettazione concettuale (6 punti)

Il sito LaSaiLaRisposta.org (LSLR) gestisce domande e risposte dei propri iscritti. Ogni utente è identificato univocamente dal suo username, e ne vengono memorizzati anche nome, cognome, indirizzo email, password per l'accesso al sito e data di iscrizione.

Ogni utente può porre delle domande, ognuna caratterizzata da un titolo, un testo e la data di pubblicazione. Ogni domanda deve fare riferimento a uno (e uno solo) degli argomenti generali previsti da LSLR. Ogni risposta fornita da un utente a una domanda ha testo e data, e può essere valutata in una scala da 1 a 5 (più o meno soddisfacente) dall'utente che ha formulato la domanda. Uno stesso utente può fornire più risposte alla stessa domanda. Quando l'utente che ha posto la domanda si ritiene soddisfatto delle risposte ricevute può bloccare ulteriori risposte, scegliendo tra le risposte ricevute una come migliore delle altre.



Commenti:

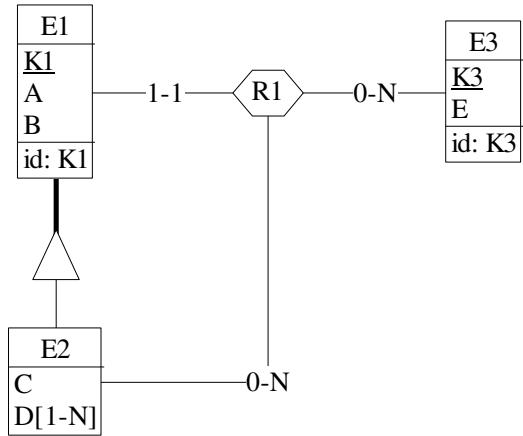
- L'esercizio è semplice, in quanto la definizione di entità e associazioni discende direttamente dalle specifiche.
- I seguenti vincoli non sono rappresentabili:
  - L'utente che ha formulato la domanda non può fornire risposte a quella domanda;
  - La risposta scelta per una domanda chiusa deve far parte dell'insieme di risposte date a quella domanda. Si noti che, se si fosse aggiunto un attributo Booleano "Migliore" all'associazione DR (anziché introdurre l'associazione SCELTE), il vincolo non esprimibile sarebbe stato quello di avere, per ogni domanda, il valore Vero al massimo per una risposta;
  - La Data di ogni risposta non può essere minore della Data della corrispondente domanda.

#### 4) Progettazione logica (6 punti totali)

Dato lo schema concettuale in figura e considerando che:

- a) tutti gli attributi sono di tipo INT;
  - b) le entità E1 ed E2 vengono tradotte insieme;
  - c) l'associazione R1 non viene tradotta separatamente;
  - d) vale la dipendenza funzionale  $A \rightarrow E$ ;

**4.1) [3 p.]** Si progettino gli opportuni schemi relazionali e si definiscano tali schemi in DB2 (sul database SIT\_STUD) mediante un file di script denominato **SCHEMI.txt**



```
CREATE TABLE E3 (  
  K3 INT NOT NULL PRIMARY KEY,  
  E INT NOT NULL
```

```
CREATE TABLE E12 (
  K1 INT NOT NULL PRIMARY KEY,
  A INT NOT NULL,
  B INT NOT NULL,
  K1E2 INT NOT NULL REFERENCES E1,
  K3 INT NOT NULL REFERENCES E3,
  TIPO SMALLINT NOT NULL CHECK (TIPO IN (1,2)),          -- 2: istanza anche di E2
  C INT,
  CONSTRAINT E2 CHECK ((TIPO = 1 AND C IS NULL) OR (TIPO = 2 AND C IS NOT NULL) );
```

```
CREATE TABLE E2D (
  K1 INT NOT NULL REFERENCES E1,
  D INT NOT NULL,
  CONSTRAINT PK PRIMARY KEY (K1,D);
```

**4.2) [3 p.]** Per i vincoli non esprimibili a livello di schema si predispongano opportuni **trigger che evitino inserimenti di tuple non corrette**, definiti in un file **TRIGGER.txt** e usando se necessario il simbolo '@' per terminare gli statement SQL (altrimenti ';')

```
-- Trigger che garantisce che una tupla di E12 referenzi, tramite K1E2, un'istanza di E2
-- Analogo trigger per la foreign key di E2D, ottenibile come da commenti
CREATE TRIGGER R1_E2
  BEFORE INSERT ON E12
  REFERENCING NEW AS N
  FOR EACH ROW
  WHEN (NOT EXISTS ( SELECT * FROM E12
    WHERE  N.K1E2 = E12.K1
    AND    E12.TIPO = 2 ) ) )
  SIGNAL SQLSTATE '70001' (La tupla inserita deve referenziare una istanza di E2!); -- resp. 70001
  -- rispettivamente E2D_E2
  -- resp. E2D
```

-- Trigger che garantisce il rispetto del vincolo al punto d)

```

CREATE TRIGGER PUNTO_D
BEFORE INSERT ON E12
REFERENCING NEW AS N
FOR EACH ROW
WHEN (EXISTS (    SELECT * FROM E12, E3 E3X, E3 E3Y
                    WHERE E12.A = N.A                      -- stesso valore di A...
                    AND E12.K3 = E3X.K3 AND N.K3 = E3Y.K3
                    AND E3X.E <> E3Y.E )                  -- ...ma diverso valore
SIGNAL SQLSTATE '70003' ('La tupla inserita non rispetta la FD da A ad E!');


```

-- Poiche' D non ammette valori nulli, ogni volta che si inserisce una tupla in E12 con TIPO = 2 bisogna anche eseguire, nella stessa transazione, almeno un inserimento in E2D.