

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

---

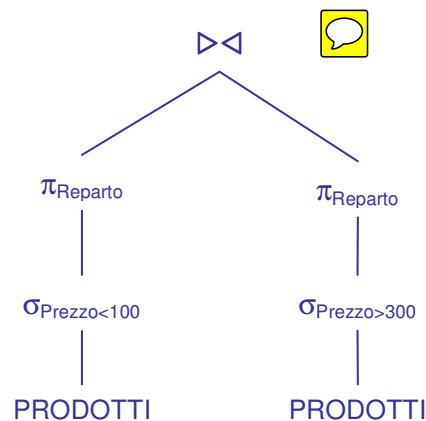
**1) Algebra relazionale (3 punti totali):**

Date le seguenti relazioni:

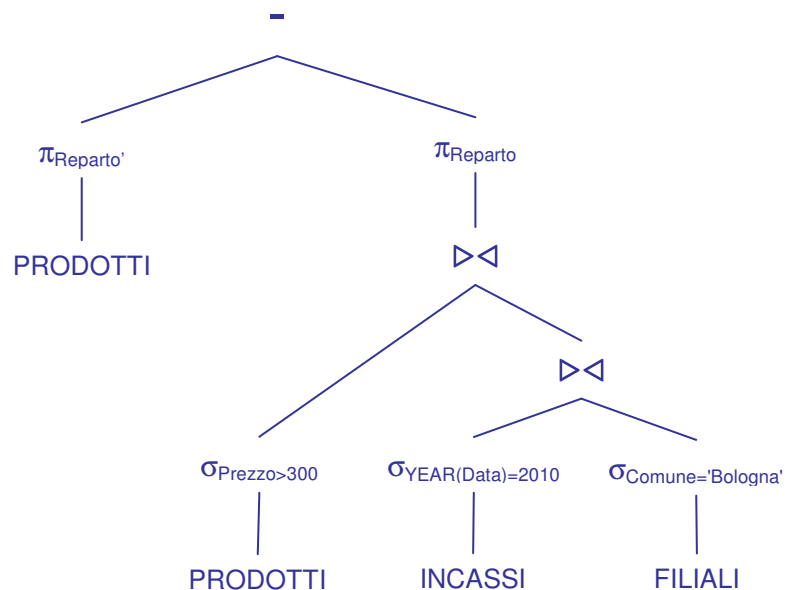
```
FILIALI (CodF, Indirizzo, Comune);  
PRODOTTI (CodP, Reparto, Prezzo);  
INCASSI (CodF, CodP, Data, Importo),  
CodF REFERENCES FILIALI, CodP REFERENCES PRODOTTI;  
-- Importo e Prezzo sono interi > 0
```

si scrivano in algebra relazionale le seguenti interrogazioni:

- 1.1) [1 p.]** Tutti i reparti che hanno in vendita **almeno un prodotto con prezzo inferiore a 100 €** e **uno con prezzo superiore a 300 €**



- 1.2) [2 p.]** I reparti che nel 2010 non hanno venduto nessun prodotto di prezzo superiore a 300 € in nessuna filiale di Bologna



## Sistemi Informativi T

19 luglio 2011

### Risoluzione

#### 2) SQL (5 punti totali)

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

**2.1) [2 p.]** I reparti che nel 2010 non hanno venduto nessun prodotto di prezzo superiore a 300 € in nessuna filiale di Bologna

```
SELECT  DISTINCT P.Reparto
FROM    PRODOTTI P
WHERE   P.Reparto NOT IN (SELECT P1.Reparto
                           FROM    PRODOTTI P1, INCASSI I, FILIALI F
                           WHERE   P1.CodP = I.CodP
                           AND     I.CodF = F.CodF
                           AND     F.Comune = 'Bologna'
                           AND     YEAR(I.Data) = 2010
                           AND     P1.Prezzo > 300)
```

**2.2) [3 p.]** Per ogni prodotto di prezzo superiore a 100 €, la filiale che ha totalizzato il maggiore incasso complessivo nel 2010

```
WITH
TOTINCASSI(CodF, CodP, TotImporto) AS (
  SELECT I.CodF, P.CodP, SUM(I.Importo)
  FROM   PRODOTTI P, INCASSI I
  WHERE  P.CodP = I.CodP
  AND    P.Prezzo > 100
  AND    YEAR(I.Data) = 2010
  GROUP BY I.CodF, P.CodP )
SELECT T.CodP, F.*, T.TotImporto
FROM   TOTINCASSI T, FILIALI F
WHERE  F.CodF = T.CodF
AND    T.TotImporto = ( SELECT MAX(T1.TotImporto)
                       FROM TOTINCASSI T1
                       WHERE T1.CodP = T.CodP )

-- La Common Table Expression calcola, per ogni prodotto di prezzo
-- superiore a 100 € e per ogni filiale, l'incasso complessivo nel 2010.
-- Il blocco SELECT confronta quindi gli incassi a parita' di prodotto
-- e restituisce, per ogni prodotto, la filiale con il maggiore incasso
```

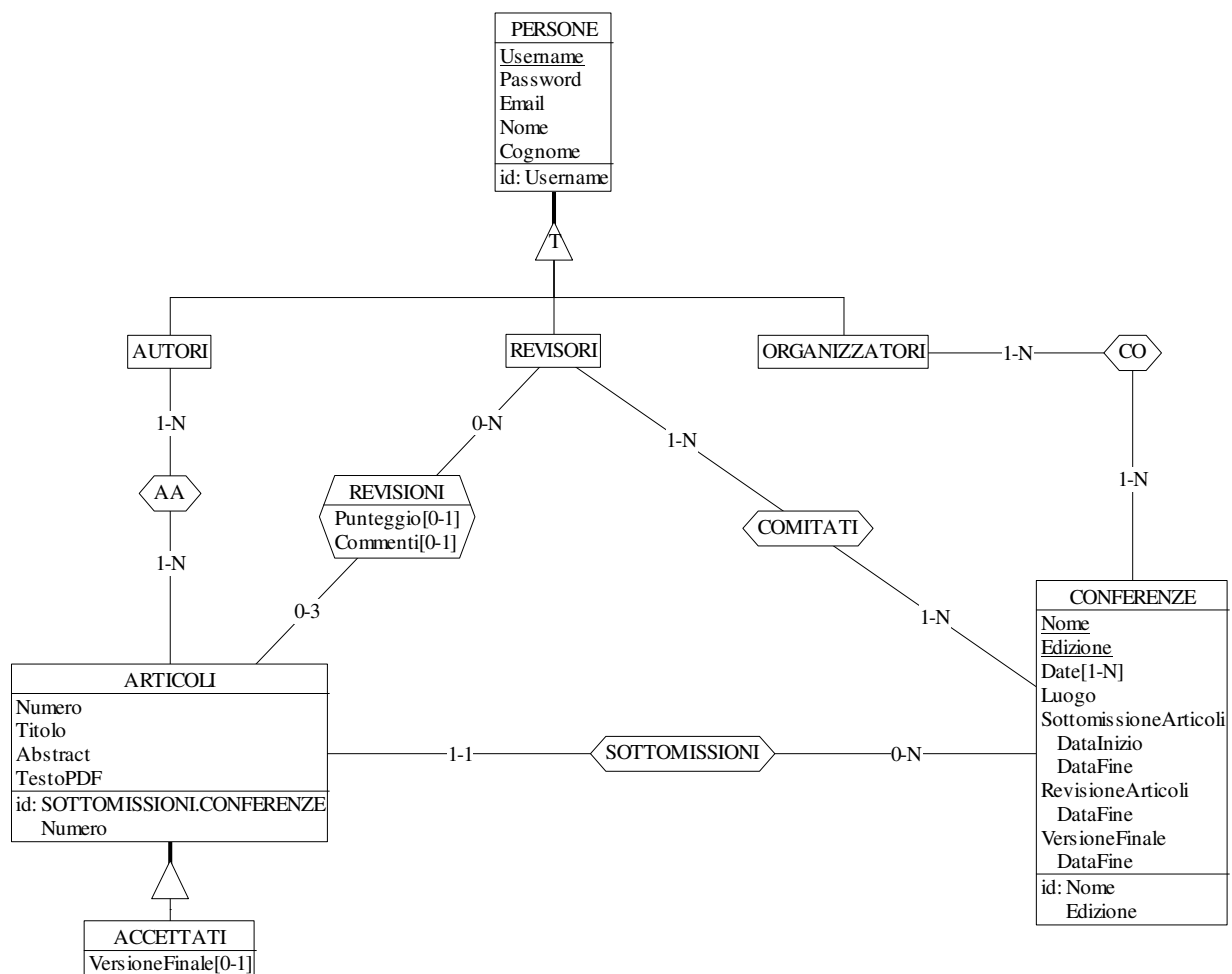
## 3) Progettazione concettuale (6 punti)

Il sito ConferenceManager (ConfMan) permette di gestire tutte le attività principali relative all'organizzazione scientifica di conferenze. Di ogni articolo scientifico sottomesso a una data conferenza, ConfMan tiene traccia degli autori (uno o più), del titolo, di un suo riassunto (*abstract*) e del file in formato PDF contenente il testo dell'articolo.

Al termine del periodo in cui è possibile sottomettere articoli, i responsabili della conferenza assegnano ogni articolo a tre revisori, selezionati tra quelli facenti parte del Comitato Scientifico della conferenza. Ogni revisore, nel periodo di tempo stabilito per quella conferenza, fornisce una valutazione dell'articolo, costituita da un punteggio e da una revisione testuale.

Al termine della fase di valutazione, il Comitato Scientifico decide quali articoli devono essere accettati e quali no. Per ogni articolo accettato è prevista la sottomissione, entro una data prefissata, di una versione finale, che tenga conto dei commenti dei revisori.

Oltre alle informazioni già citate, di ogni conferenza ConfMan mantiene i dati sul nome della conferenza e sulla sua edizione (ad es. International Conference on Conceptual Modeling, Sixth edition), che congiuntamente identificano una conferenza, sul periodo e sul luogo di svolgimento.



## Commenti:

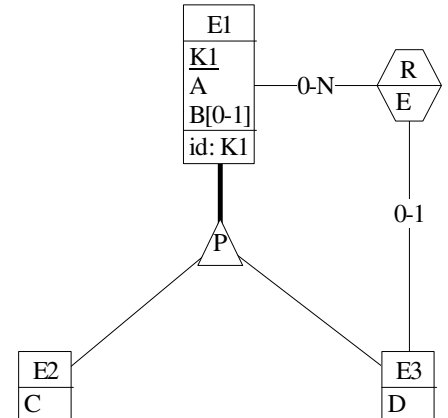
- La gerarchia è solo totale, in quanto non vi è incompatibilità dichiarata tra i vari ruoli
- Alcune cardinalità minime sono poste a 0, in quanto è necessario considerare l'evoluzione temporale dei dati (ad es.: per i lavori ACCETTATI la versione finale non è subito disponibile, così come quali sono i revisori di un articolo e le loro valutazioni). Volendo lo stesso si poteva fare per il Comitato di una Conferenza (se si suppone che quando si inserisce una nuova Conferenza non sia necessario specificare subito la composizione del Comitato)
- Non è possibile rappresentare il vincolo che i revisori di un articolo facciano effettivamente parte del Comitato della Conferenza a cui l'articolo è stato sottomesso

**Sistemi Informativi T**  
**19 luglio 2011**  
**Risoluzione**

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

Dato lo schema concettuale in figura  
e considerando che:

- a) tutti gli attributi sono di tipo INT;
- b) l'associazione R non viene tradotta separatamente;
- c) le entità E1 ed E2 vengono tradotte assieme, e separatamente dall'entità E3;
- d) un'istanza di E3 non è mai associata, tramite R, a se stessa;



**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 E1E2(  
  K1 INT NOT NULL PRIMARY KEY,  
  A INT NOT NULL,  
  B INT,  
  TIPO SMALLINT NOT NULL CHECK (TIPO IN (2,3)),      -- 2: istanza di E2; 3: istanza di E3  
  C INT,  
  CONSTRAINT E2E3 CHECK (  
    (TIPO = 2 AND C IS NOT NULL) OR (TIPO = 3 AND C IS NULL))  );
```

```
CREATE TABLE E3(  
  K1 INT NOT NULL PRIMARY KEY REFERENCES E1E2,  
  D INT NOT NULL,  
  K1R INT REFERENCES E1,  
  E INT,  
  CONSTRAINT R_E_PUNTO_D CHECK ( (K1R IS NOT NULL AND E IS NOT NULL AND K1R <> K1)  
    OR (K1R IS NULL AND E IS NULL) );
```

**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 il simbolo '@' per terminare gli statement SQL

```
-- Quando si inserisce una tupla in E1E2 di TIPO = 3 (ovvero di E3), bisogna anche eseguire, nella stessa  
-- transazione, un inserimento in E3  
-- Il seguente trigger garantisce semplicemente che il valore di TIPO della tupla appena inserita in E1E2 sia corretto
```

```
CREATE TRIGGER INSERT_E3  
NO CASCADE BEFORE INSERT ON E3  
REFERENCING NEW AS N  
FOR EACH ROW MODE DB2SQL  
WHEN (NOT EXISTS (SELECT *  
                  FROM E1E2  
                  WHERE E1E2.K1 = N.K1  
                  AND E1E2.TIPO = 3 ))  
SIGNAL SQLSTATE '70001' ('La tupla inserita in E1E2 non ha il valore di TIPO giusto!')@
```