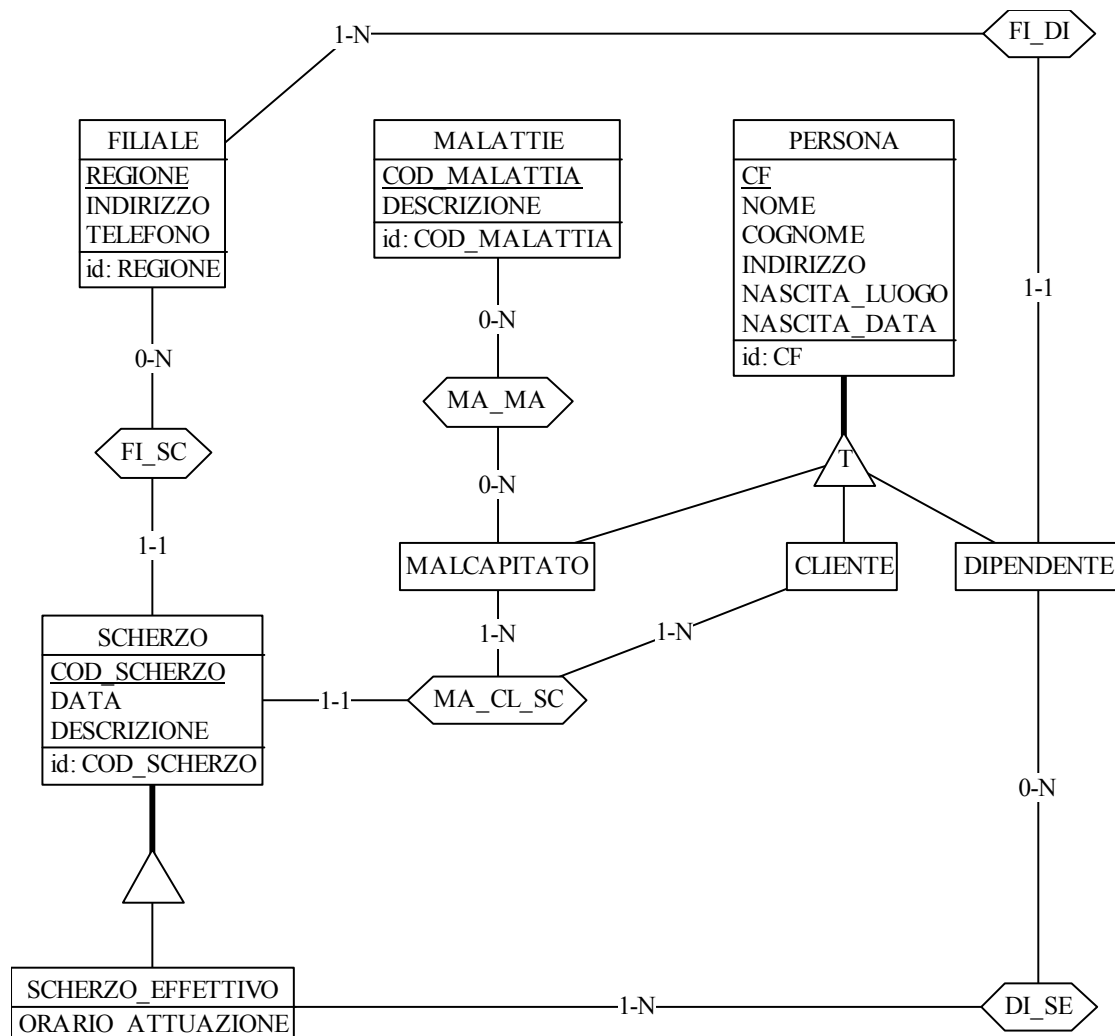


Sistemi Informativi L-B - Prova integrata
20 gennaio 2006 – Compito 2
Risoluzione

Tempo a disposizione: 2 ore

1) Progettazione concettuale (5 punti)



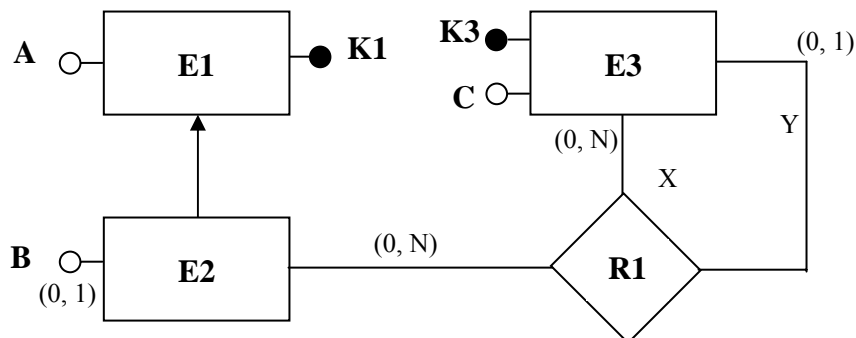
Commenti:

- La copertura della gerarchia che ha come entità padre PERSONA è di tipo totale e sovrapposto; ciò permette di modellare il fatto che i dipendenti possono eseguire e ricevere scherzi (oltre al caso più generale che chiunque esegua uno scherzo possa anche riceverlo e viceversa)
- Le filiali (una o più) contattate da un cliente sono rilevabili tramite le associazioni MA_CL_SC e FI_SC
- Si noti che non è possibile specificare il vincolo che gli addetti assegnati all'esecuzione di uno scherzo appartengano effettivamente alla filiale che ha preso in gestione lo scherzo stesso

Sistemi Informativi L-B - Prova integrata
20 gennaio 2006 – Compito 2
Risoluzione

2) Progettazione logica e normalizzazione (3 punti)

Dato lo schema concettuale in figura



e considerando che:

- a) tutti gli attributi sono di tipo CHAR(10);
- b) le entità E1 ed E2 vengono accorpate;
- c) l'associazione R1 NON viene tradotta separatamente
- d) l'associazione R1 NON mette mai in relazione un'istanza di E3 con se stessa

si progettino gli opportuni schemi relazionali e si definiscano **nel database STUDENTI** tali schemi facendo uso dell'SQL di DB2; per gli eventuali vincoli non esprimibili a livello di schema si predispongano opportune **query di verifica da eseguire prima di effettuare inserimenti di tuple**, allo scopo di evitare che tali inserimenti violino i vincoli stessi.

```
CREATE TABLE E1 (  
    K1 CHAR(10) NOT NULL PRIMARY KEY,  
    A CHAR(10) NOT NULL,  
    B CHAR(10),  
    SEL SMALLINT NOT NULL CHECK (SEL IN (1, 2)),          -- 1: E1; 2: E2  
    CONSTRAINT CHECK_HIERARCHY  
        CHECK ( SEL=2 OR (SEL=1 AND B IS NULL) )  
);  
  
CREATE TABLE E3 (  
    K3 CHAR(10) NOT NULL PRIMARY KEY,  
    C CHAR(10) NOT NULL,  
    KR1X CHAR(10) REFERENCES E3,  
    KR1E2 CHAR(10) REFERENCES E1,  
    CONSTRAINT CHECK_AVOID_SELF_REF  
        CHECK ( KR1X <> K3 )  
);
```

Query che, data la tupla ('k3', 'c', 'kr1x', 'kr1e2') che si intende inserire in E3, verifica che la tupla di E1 che viene referenziata sia una istanza di E2:

```
SELECT * FROM E1 WHERE K1 = 'kr1e2' AND SEL=2;          -- ok se restituisce una tupla
```

3) Transazioni (2 punti)

In caso di system failure un DBMS può evitare di effettuare riscritture inutili operando come segue:

- quando una pagina P è modificata da una transazione T, viene generato il Log record relativo e gli viene assegnato un log sequence number (LSN)
- il LSN viene quindi scritto nel page header di P
- quando T viene rifatta e si trova un Log record relativo a P con LSN = k, se vale $LSN(P) \geq k$ allora non è necessario riscrivere P

In questo modo si leggono tutte le pagine modificate da T, ma si riscrivono solo quelle strettamente necessarie.