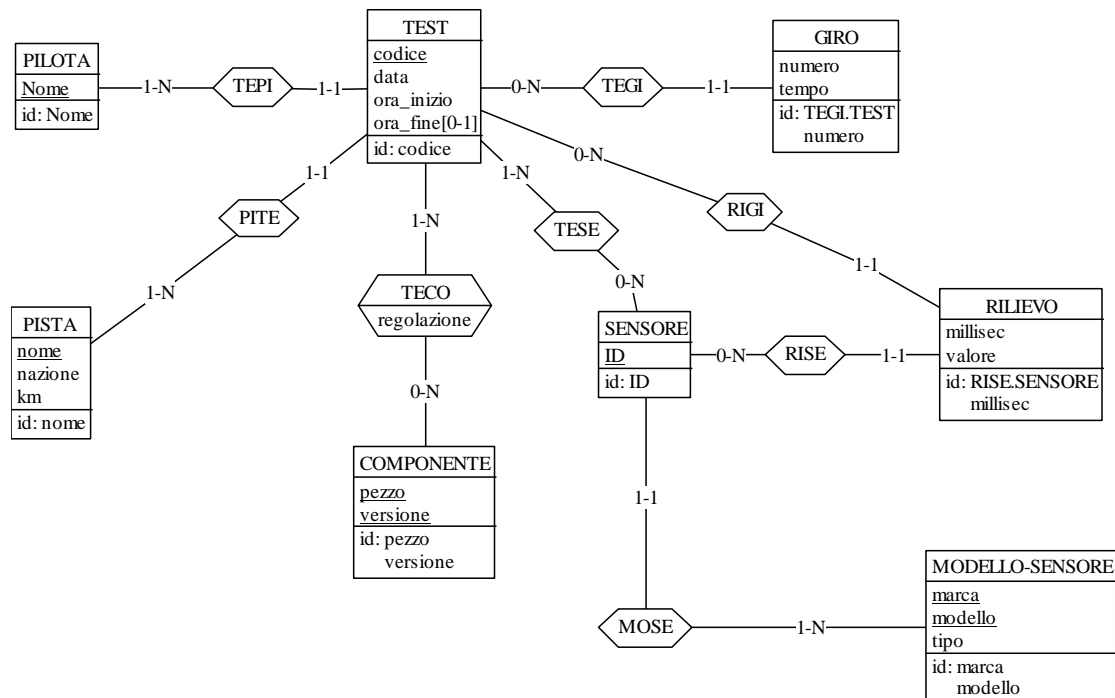


**Tempo a disposizione: 2 ore**

**1) Progettazione concettuale (5 punti)**

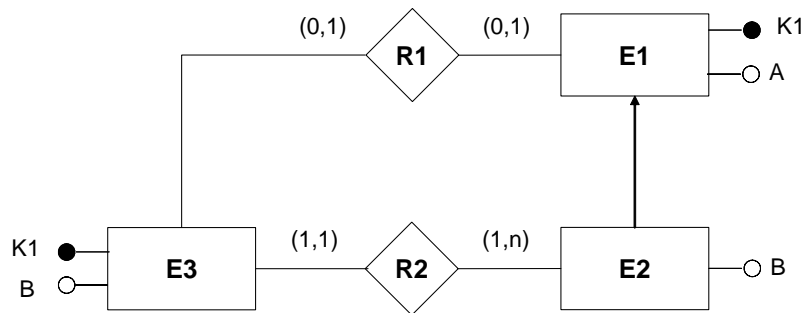


**Commenti:**

- L'entità **SENSORE** ha come istanze i sensori specifici utilizzati nei test, ovvero in un dato istante un sensore non può eseguire più di un rilievo. Da questa osservazione deriva che l'identificatore di **RILIEVO** non necessita del codice del **TEST**.
- Nella soluzione si è supposto che il DB sia alimentato con nuovi dati man mano che un test evolve. Da questa ipotesi deriva che **TEST.ora\_fine** è parziale (non definito sino alla fine del test) e che le cardinalità minime di **TEST** per le associazioni **TEGI** e **RIGI** sono 0.

## 2) Progettazione logica e normalizzazione (3 punti)

Dato lo schema concettuale in figura



e considerando che:

- a) tutti gli attributi sono di tipo INT;
- b) E2 viene tradotta con E1;
- c) le associazioni R1 e R2 vengono tradotte con E3;

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 INT NOT NULL PRIMARY KEY,
    A INT NOT NULL,
    B INT
);

CREATE TABLE E3 (
    K1 INT NOT NULL PRIMARY KEY,
    B INT NOT NULL,
    K1R1 INT UNIQUE REFERENCES E1,      -- traduce R1
    K1R2 INT NOT NULL REFERENCES E1    -- traduce R2
);
```

L'inserimento di una tupla (k1,b,k1r1,k1r2) in E3 è consentito solo se k1r2 corrisponde a una tupla di E2:

```
SELECT * FROM E1
WHERE K1 = k1r2
AND B IS NOT NULL;
```

*-- ok se restituisce una tupla*

All'atto dell'inserimento di una tupla (k1,a,b) in E1 (ovvero la tupla è un'istanza di E2) è necessario prevedere, a causa della cardinalità minima 1 di R2 per E2, un inserimento di tipo transazionale con almeno una tupla di E3 per cui E3.K1R2 = k1.

## 3) Elaborazione delle interrogazioni (2 punti)

Si vedano le dispense del corso per la definizione di valutazione in pipeline di un operatore.

Un semplice piano di accesso con un operatore che lavora in pipeline e un altro no è, ad esempio:

```
Sort[<criterio di ordinamento>]
  TableScan(R)
```

in quanto il sort richiede di materializzare i dati, mentre la scansione sequenziale di una relazione no.