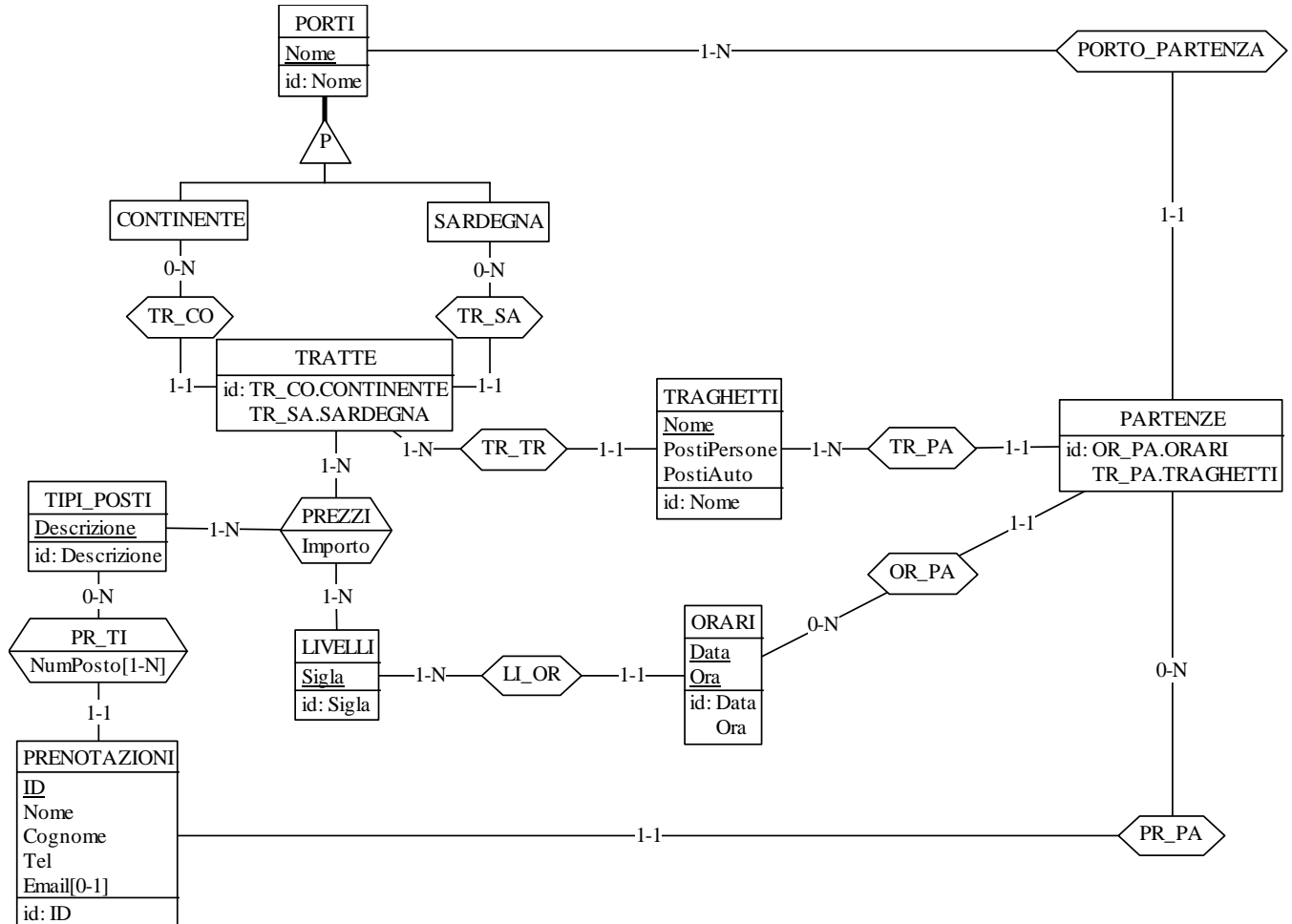


Sistemi Informativi L-B
18 giugno 2009
Risoluzione

Tempo a disposizione: 2 ore

1) Progettazione concettuale (5 punti)

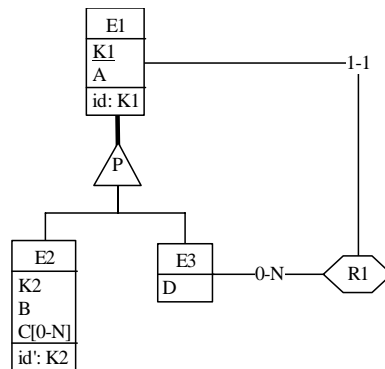


Commenti:

- La principale difficoltà dell'esercizio riguarda la corretta modellazione degli orari dei traghetti e delle relative prenotazioni. La soluzione proposta si caratterizza per i seguenti elementi:
- La presenza di una serie di entità "statiche" di supporto, quali TIPI_POSTI, LIVELLI e ORARI. Quest'ultima si rende necessaria per associare a ciascuna partenza un ben determinato livello (associare direttamente PARTENZE a LIVELLI non sarebbe corretto, in quanto i livelli non dipendono dai traghetti ma solo da data e ora)
- La presenza di un'entità TRATTE, che non distingue tra porto di partenza e di arrivo. Questa entità permette di rappresentare correttamente le specifiche sui traghetti (ogni traghetto opera su una singola tratta) e sui prezzi (i prezzi dipendono anche dalla tratta)
- Le PARTENZE sono univocamente definite da un traghetto e da un orario. La reificazione di PARTENZE (che altrimenti si sarebbe potuta rappresentare come un'associazione tra ORARI e TRAGHETTI) si rende necessaria a causa delle associazioni con PRENOTAZIONI e PORTI. Quest'ultima ha lo scopo di esplicitare qual è il porto di partenza
- Ogni prenotazione ha come proprietà i dati del cliente, una partenza (associazione PR_PA) e un tipo di posto (associazione PR_TI). I biglietti sono rappresentati nell'associazione PR_TI mediante un attributo multi-valore.
- L'esercizio è un ulteriore esempio del "pattern" libro-copia di libro, in cui la corretta soluzione richiede di distinguere tra aspetti statici e dinamici. In questo caso la coppia di entità che realizza il pattern è TRAGHETTI-PARTENZE

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) la gerarchia viene collassata verso il basso;
- c) l'associazione R1 non viene tradotta separatamente;
- d) un'istanza di E3 non è mai associata tramite R1 a istanze di E1 con A=10;

si progettino gli opportuni schemi relazionali e si definiscano 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 E2 (
  K1 INT NOT NULL PRIMARY KEY,
  A INT NOT NULL CONSTRAINT Punto_d CHECK (A <> 10),
  K2 INT NOT NULL UNIQUE,
  B INT NOT NULL,
  K1R1 INT NOT NULL REFERENCES E3 );
```

```
CREATE TABLE E3 (
  K1 INT NOT NULL PRIMARY KEY,
  A INT NOT NULL CONSTRAINT Punto_d CHECK (A <> 10),
  D INT NOT NULL
  K1R1 INT NOT NULL REFERENCES E3 );
```

```
CREATE TABLE E2C (
  K1 INT NOT NULL REFERENCES E2,
  C INT NOT NULL,
  PRIMARY KEY (K1,C) );
```

Per garantire che un valore di K1 non sia presente contemporaneamente in E2 ed E3, all'atto dell'inserimento di una tupla (k1, ...) in E2 (analogamente quando si inserisce in E3) va eseguita la query:

```
SELECT * FROM E3 -- ok se non restituisce nessuna tupla
WHERE K1 = k1;
```

Per quanto riguarda il vincolo al punto d), si noti che esso di fatto impedisce istanze di E1 in cui A=10. Se infatti vi fosse un'istanza con A=10, questa sarebbe necessariamente associata a un'istanza di E3 e il vincolo sarebbe violato. E' quindi sufficiente inserire nelle due relazioni E2 ed E3 la clausola CHECK (A<>10).

3) DB fisico (2 punti)

Si vedano le slides del corso "DB fisico", pp. 24-26.