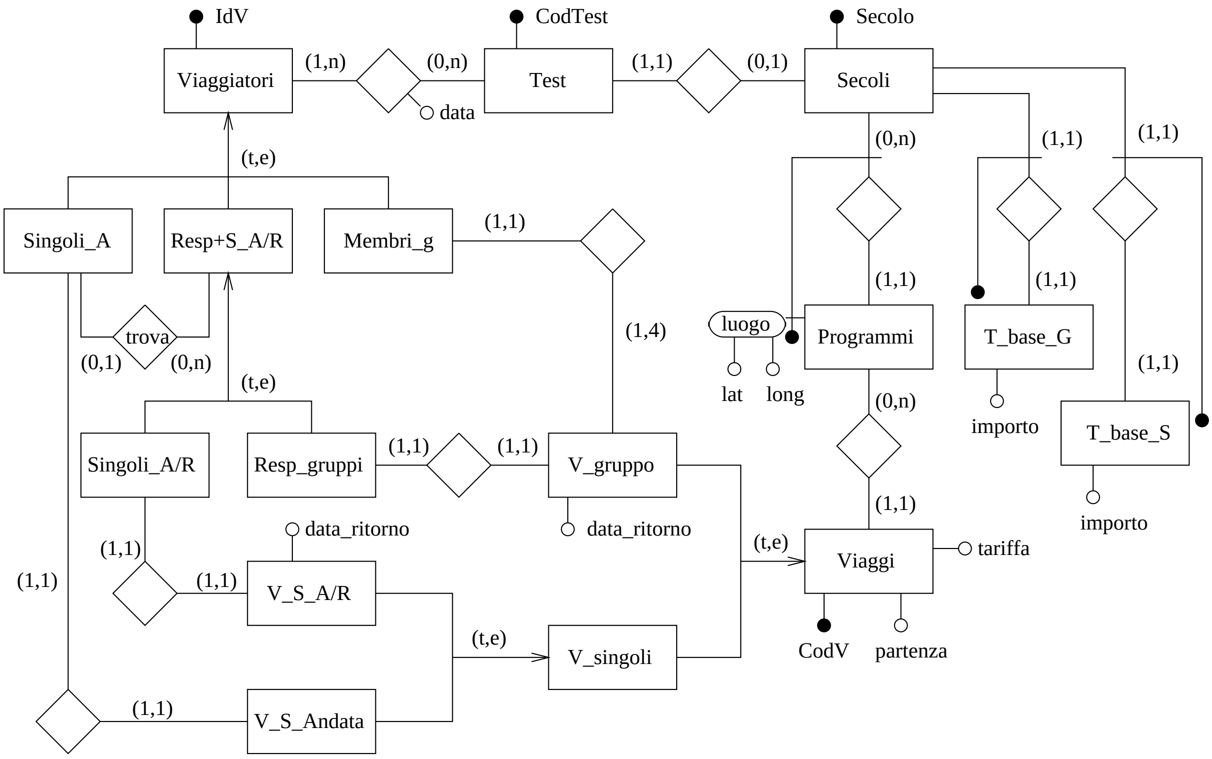
Corso di Laurea in Informatica
Basi di Dati e Sistemi Informativi – 16 Giugno 1997
Risoluzione

Esercizio 1



La complessità dell'esercizio è principalmente dovuta alla modellazione non banale della gerarchia relativa ai `Viaggiatori`. L'analisi del testo, e in particolare del terzo paragrafo:

> *Viaggi di sola andata, possibili solo per singoli, possono comunque essere interrotti, gratuitamente, se il viaggiatore riesce a trovare qualcuno con il biglietto di ritorno, comunicando il nome della persona con biglietto (nome del responsabile se si tratta di un gruppo) alla sede della T^3.*

porta a partizionare l'entità `Viaggiatori` come mostrato, e a introdurre l'associazione `trova`.

La gerarchia relativa ai `Viaggi` si struttura più naturalmente come in figura. Si noti che non esiste un'entità `Gruppi`, ma la composizione di un gruppo è ricavabile dalle associazioni che interessano l'entità `V_gruppo` (viaggi di gruppo).

Esercizio 2

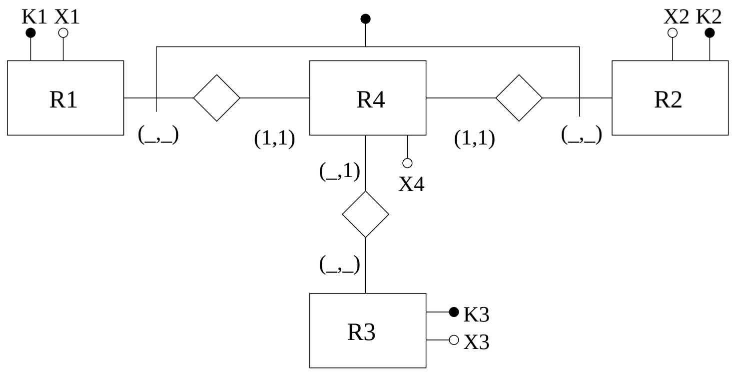
Per eseguire correttamente l'inserimento della tupla `('r','s')` occorre verificare che:

1. la tupla `('r','s')` non sia già presente nella relazione `Resp_Settori` (vincolo di *primary key*);
2. esista una tupla con valore di chiave `'r'` nella relazione `Resp_Citta` (vincolo di *integrità referenziale*);
3. il settore `'s'` della città di `'r'` non abbia già un responsabile (vincolo $CS \rightarrow R$).

Per i primi due punti la soluzione è immediata e viene qui omessa. Per il punto 3 si può fare uso della seguente istruzione SQL e procedere all'inserimento solo se il risultato è 0:

```
Select count(*)
From   Resp_Settori S, Resp_Citta C
Where  S.Responsabile = C.Responsabile
and    S.Settore = 's'
and    C.Citta in ( Select Citta
                    From   Resp_Citta
                    Where  Responsabile = 'r' );
```

Esercizio 3



Lo schema E/R (in cui le cardinalità non soggette a vincoli sono rappresentate con il simbolo '_') presenta, come unica particolarità, un meccanismo di identificazione dell'entità R_4 tale da giustificare la presenza dello schema $R_4(\underline{K_1, K_2}, K_3, X_4)$. Quest'ultimo si ottiene dallo schema E/R importando gli identificatori esterni (K_1 e K_2) e traducendo l'associazione tra R_4 e R_3 mediante inglobamento dell'identificatore di R_3. Perché ciò sia possibile è evidentemente richiesto che R_4 partecipi con cardinalità massima 1 a tale associazione.

Esercizio 4

Si osservi innanzitutto che un piano di accesso è univocamente rappresentato mediante una corrispondente *permutazione* delle relazioni R_1, R_2 e R_3. Ne segue che i piani di accesso da considerare sono in numero pari a $3! = 6$.

Per le ipotesi fatte, le cardinalità dei join/prodotti Cartesiani sono pari a:

$$
\begin{aligned}
\mid R_1 \bowtie R_2 \mid &= NT_{R_2} \\
\mid R_2 \bowtie R_3 \mid &= NT_{R_2} \\
\mid R_1 \bowtie R_3 \mid &= NT_{R_1} \times NT_{R_3}
\end{aligned}
$$

Indicando per semplicità un piano di accesso con la sequenza ordinata dei pedici che identificano le 3 relazioni, si ha la seguente tabella dei costi, che risolve pertanto la prima parte dell'esercizio:

Piano di accesso	Costo
$(1, 2, 3)$	$NT_{R_1} + NT_{R_2}$
$(1, 3, 2)$	$NT_{R_1} + NT_{R_1} \times NT_{R_3}$
$(2, 1, 3)$	$NT_{R_2} + NT_{R_2}$
$(2, 3, 1)$	$NT_{R_2} + NT_{R_2}$
$(3, 1, 2)$	$NT_{R_3} + NT_{R_1} \times NT_{R_3}$
$(3, 2, 1)$	$NT_{R_3} + NT_{R_2}$

I piani che eseguono il prodotto Cartesiano tra R_1 e R_3 sono $(1, 3, 2)$ e $(3, 1, 2)$, e possono essere ottimali solo se $NT_{R_1} \times NT_{R_3} < NT_{R_2}$. Se ciò vale, si deriva immediatamente che $NT_{R_1} < NT_{R_2}$ e $NT_{R_3} < NT_{R_2}$, che garantiscono che uno dei due piani è sicuramente ottimale. In particolare, se $NT_{R_1} < NT_{R_3}$ allora il piano $(1, 3, 2)$ è ottimale. Viceversa, quando $NT_{R_3} < NT_{R_1}$, è il piano $(3, 1, 2)$ ad essere ottimale.

Esercizio 5 (facoltativo)

La soluzione è data dalla seguente istruzione SQL:

```
Select C.Citta
From   Resp_Citta C
Where  C.Citta not in ( Select C1.Citta
                        From   Resp_Citta C1
                        Where  C1.Responsabile in ( Select   Responsabile
                                                    From     Resp_Settori
                                                    Group by Responsabile
                                                    Having   count(*) > 1) );
```