

## SOLUZIONI

### Esercizio 1

- $\sigma_F(E_1 - E_2) = \sigma_F(E_1) - \sigma_F(E_2)$ . V
- $\pi_X(E_1 - E_2) = \pi_X(E_1) - \pi_X(E_2)$ . F
- ✓ • I comandi SQL **delete from T** e **drop table T** hanno lo stesso risultato nel caso la tabella T sia vuota (cioè non contenga alcuna tupla). F
- • Il calcolo relazionale è un linguaggio dichiarativo in cui le espressioni descrivono le proprietà del risultato. V
- • Il linguaggio SQL esprime le interrogazioni in modo procedurale. F

### Esercizio 2

Si considerino le seguenti relazioni. Ogni treno effettua ogni giorno la stessa tratta (esattamente una).

Treno(IDTreno, num\_vagoni, num\_posti)

Tratta(IDTreno, Partenza, Arrivo, km)

Viaggi(IDCliente, IDTreno, Data)

a) Per ogni possibile destinazione, ricavare il numero minimo di chilometri necessari per arrivarvi da Bologna senza cambiare (cioè utilizzando un solo treno), in SQL.

**soluzione**

```
select Arrivo, min(km)
from Tratta
where Partenza='Bologna'
group by Arrivo;
```

b) Ricavare il numero minimo di chilometri necessari per viaggiare da Bologna a Napoli effettuando esattamente un cambio (cioè utilizzando due treni), in SQL.

**soluzione**

```
select min(T1.km+T2.km)
from Tratta T1 join Tratta T2 on T1.Arrivo = T2.Partenza
where T1.Partenza = 'Bologna' And T2.Arrivo = 'Napoli';
```

c) Elencare gli ID dei clienti che il 3 gennaio 2005 hanno viaggiato da Bologna a Napoli effettuando esattamente un cambio (cioè utilizzando due treni), in Algebra Relazionale (selezione, proiezione, natural join, ridenominazione).

CALCOLO RELAZIONALE  
MODO PROCEDURALE

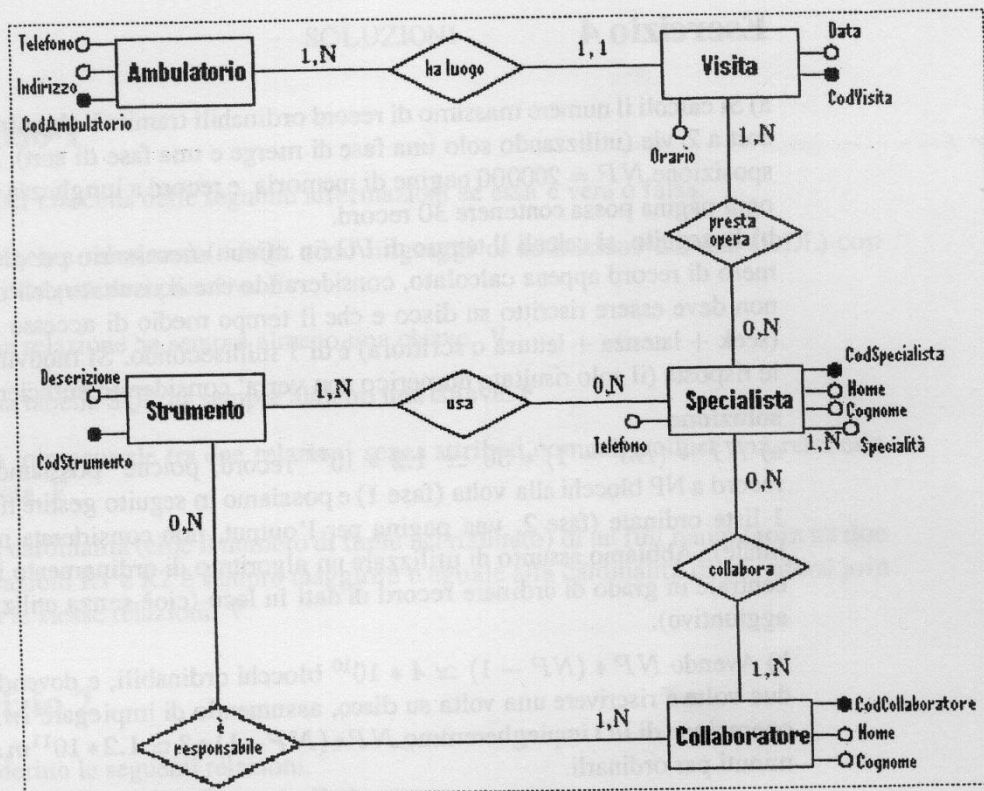


### soluzione

$\pi_{IDCliente}(\sigma_{Partenza='Bologna' \wedge Arrivo='Napoli'}(\rho_{Treno1,Cambio,km1 \leftarrow IDTreno,Arrivo,km(Tratta)} \bowtie \rho_{Treno2,Cambio,km2 \leftarrow IDTreno,Partenza,km(Tratta)})) \bowtie (\rho_{Treno1 \leftarrow IDTreno(Viaggi)} \bowtie \rho_{Treno2 \leftarrow IDTreno(\sigma_{Data='03-01-2005'}(Viaggi))))$

### Esercizio 3

- Si vuole modellare un sistema per la gestione di visite mediche da svolgersi in diversi ambulatori medici.
- Ogni visita è svolta in uno ed un solo ambulatorio avente un codice univoco e caratterizzato da indirizzo e numero di telefono.
- In un ambulatorio possono essere svolte una o più visite.
- Ogni visita è svolta in una precisa data ed orario oltre ad avere un codice univoco che la caratterizza.
- Ogni visita può richiedere la prestazione d'opera di uno o più specialisti.
- Ogni specialista può svolgere più visite.
- Ogni specialista è identificato da un codice univoco e prevede l'indicazione delle sue specialità, nome, cognome e recapito telefonico.
- Ogni specialista può avere necessità di un certo insieme di strumenti per i quali è presente un codice univoco ed una descrizione.
- Uno strumento può essere utilizzato da uno o più specialisti.
- Ogni specialista può avere bisogno di più collaboratori caratterizzati da un codice univoco, nome e cognome.
- Un collaboratore collabora con almeno uno specialista, e può essere responsabile dell'utilizzo di uno o più strumenti.
- Uno strumento può essere utilizzato da più collaboratori (ma anche da nessuno).
- Si disegni uno schema E/R che modelli la realtà descritta, integrandolo se necessario con business rules.



19



## Esercizio 4

a) Si calcoli il numero massimo di record ordinabili tramite l'algoritmo di merge-sort a Z vie (utilizzando solo una fase di merge e una fase di sort), avendo a disposizione  $NP = 200000$  pagine di memoria, e record a lunghezza fissa tali che ogni pagina possa contenere 30 record.

b) In seguito, si calcoli il tempo di I/O (in minuti) necessario per ordinare il numero di record appena calcolato, considerando che il risultato dell'ordinamento non deve essere riscritto su disco e che il tempo medio di accesso a un blocco (seek + latenza + lettura o scrittura) è di 1 millisecondo. Si motivino entrambe le risposte (il solo risultato numerico non verrebbe considerato sufficiente).

### soluzione

a)  $NP * (NP - 1) * 30 \simeq 1.2 * 10^{12}$  record, poiché possiamo ordinare i record a NP blocchi alla volta (fase 1) e possiamo in seguito gestire fino a  $NP - 1$  liste ordinate (fase 2, una pagina per l'output, non considerata nel risultato finale). Abbiamo assunto di utilizzare un algoritmo di ordinamento in memoria centrale in grado di ordinare record di dati in loco (cioè senza utilizzare spazio aggiuntivo).

b) Avendo  $NP * (NP - 1) \simeq 4 * 10^{10}$  blocchi ordinabili, e dovendoli leggere due volte e riscrivere una volta su disco, assumendo di impiegare  $1ms$  per ogni operazione di I/O impiegheremmo  $NP * (NP - 1) * 3 \simeq 1.2 * 10^{11} ms = 2 * 10^6$  minuti per ordinarli.

## Esercizio 5

Si descriva il rapporto tra schedulazioni seriali, serializzabili e *conflict-serializability*.

### soluzione

Una schedulazione seriale è sia serializzabile che conflict-serializable. Mentre la conflict-serializability (che è verificata in assenza di conflitti WRITE-READ, READ-WRITE e WRITE-WRITE) implica la serializzabilità, non tutte le schedulazioni serializzabili sono conflict-serializable. In particolare, ciò avviene quando un elemento del database su cui si verifica un conflitto viene sovrascritto da una transazione successiva che non presenta conflitti.