

1) Il sistema informativo dell'aeroporto di Barcellona all'atterraggio di ogni volo assegna il gate (uscita-parcheggio) cercando di minimizzare la distanza che i passeggeri devono percorrere per raggiungere le rispettive coincidenze (il prossimo volo da prendere). Si consideri la seguente porzione di database

GATE(Num, Libero)

GATE_DIST(NumGateA:GATE, NumGateB:GATE, Dist)

PASSEGGERI(Cod, Volo:VOLI, Nome, Cognome, ProssimoVolo:VOLI)

VOLI(Cod, Gate:GATE, Compagnia, Modello, OraPartenza, OraArrivo)

a) Si definisca la base di dati su ORACLE

b) Si scriva la procedura `AssegnaGate(p_volo)` che, dato il codice di un volo salva in `VOLI` il gate assegnatogli. Per calcolare il miglior gate la procedura:

1) Calcola il numero di passeggeri di `p_volo` per le diverse coincidenze.

2) Per ogni gate libero `g` calcola la somma delle distanze che i vari gruppi di passeggeri dovrebbero percorrere per raggiungere il gate della coincidenza se il volo fosse assegnato a `g`. Ovviamente la distanza deve essere moltiplicata per il corrispondente numero di passeggeri

3) Aggiorni `VOLI` inserendo il gate che minimizza tale somma

Per esempio, si supponga di assegnare al volo A il gate M5 e si supponga che sul volo vi siano 3 passeggeri che devono raggiungere il gate M7 e 2 che devono raggiungere il gate M1 ($dist(M5,M1)=4$ e $dist(M5,M7)=9$). Allora la distanza totale percorsa dai passeggeri sarà: $3 \times 9 + 2 \times 4 = 35$

2) Utilizzando il database TPCD_SRVORACLE (UID: usersi – PWD: usersi),

a) si disegni l'albero di esecuzione proposto da ORACLE per la seguente query SQL che restituisce, per ogni marca di parte, la quantità totale venduta nel 1997 a clienti francesi

```
SELECT    P_BRAND, sum(L_QUANTITY)
FROM      PART, LINEITEM, ORDERS, CUSTOMER, NATION, TIME
WHERE      N_NAME='FRANCE' AND T_YEAR=1997
            AND L_PARTKEY=P_PARTKEY AND O_ORDERKEY =L_ORDERKEY
            AND O_ORDERDATE = T_TIMEKEY AND O_CUSTKEY = C_CUSTKEY
            AND C_NATIONKEY = N_NATIONKEY
GROUP BY  P_BRAND;
```

ATTENZIONE: non lanciare la query ma utilizzare il comando EXPLAIN PLAN per verificarne il piano di esecuzione

b) Si scriva l'espressione di algebra relazionale corrispondente

c) Si mostri il piano di esecuzione ottenibile utilizzando le regole di trasformazione per espressioni equivalenti di algebra relazionale e l'algoritmo di ottimizzazione euristico studiato a lezione. Mostrare i passaggi indicando la fase dell'algoritmo utilizzata.

Attenzione si supponga che l'algoritmo consideri come soluzione dipartenza la soluzione non ottimizzata basata sul prodotto cartesiano.