

**Compito di Sistemi Informativi I**  
**(Ing. settore Informazione)**  
**del 22 luglio 2002**

**ESERCIZIO 1**

Date le relazioni:

**IMPIEGATI**

<u>MATRICOLA</u>	<u>NOME</u>	<u>RESIDENZA</u>	<u>CLASSE</u>	<u>DIPARTIMENTO</u>
YZ12	CARLO ROSSI	MODENA	III	DV1
XX21	ANNA CORLI	FERRARA	IV	DP3
YY12	MARTA VERDI	BOLOGNA	II	DV1
...	...	...	...	...

**DIPARTIMENTI**

<u>CODICE</u>	<u>DENOMINAZ</u>	<u>BUDGET</u>	<u>SEDE</u>
DP3	PERSONALE	200	ROMA
DV1	VENDITE	180	BOLOGNA
...	...	...	...

Si scriva il codice SQL che, *considerando solamente gli impiegati appartenenti alla classe più numerosa*, determini per ogni dipartimento: nome, budget e numero degli impiegati afferenti.

**ESERCIZIO 2**

Si illustrino le tecniche conosciute per il mantenimento delle proprietà **ACID** delle *transazioni* in un **DBMS distribuito**.

**ESERCIZIO 3**

Si descrivano e discutano le **strategie** che possono essere utilizzate per la *progettazione concettuale* di una base di dati utilizzando il modello Entity-Relationship.

**ESERCIZIO 4**

Date le relazioni dell'Esercizio 1, si ottimizzi l'esecuzione della seguente query:

```
SELECT * FROM IMPIEGATI, DIPARTIMENTI
WHERE DIPARTIMENTO = CODICE
      AND ( CLASSE="I" OR CLASSE="II" )
      AND RESIDENZA IN ("BOLOGNA", "FERRARA", "MODENA", "RAVENNA")
ORDER BY SEDE
```

sapendo che si ha (per i sort si assuma Z=4 vie):

IMPIEGATI: NP=10.000, NT=120.000  
DIPARTIMENTI: NP=80, NT=160  
IX(RESIDENZA): unclustered (TID ord.), NL=2.000, NK=60  
IX(DIPARTIMENTO): clustered, NL=1.500  
IX(CLASSE): unclustered (TID ord.), NL=1.800, NK=20

Si consideri infine anche il metodo **simple-hash** per l'esecuzione del join.