

# Sistemi informativi, primo modulo — 17 novembre 2000

## Compito A

Tempo a disposizione un'ora e trenta minuti. Libri chiusi.

### Soluzioni

#### Domanda 1 (15%)

Considerare una relazione  $R(A, \underline{B}, \underline{C}, D, E)$ . Indicare quali delle seguenti proiezioni hanno certamente lo stesso numero di ennuple di  $R$ :

1.  $\pi_{ABCD}(R)$  **SÌ**
2.  $\pi_{AC}(R)$  **NO**
3.  $\pi_{BC}(R)$  **SÌ**
4.  $\pi_C(R)$  **NO**
5.  $\pi_{CD}(R)$  **NO**

#### Domanda 2 (15%)

Considerare le relazioni  $R_1(\underline{A}, B, C)$  e  $R_2(\underline{D}, E, F)$  aventi rispettivamente cardinalità  $n_1$  e  $n_2$ . Assumere che sia definito un vincolo di integrità referenziale fra l'attributo  $C$  di  $R_1$  e la chiave  $D$  di  $R_2$ . Indicare la cardinalità di ciascuno dei seguenti join (indicando l'intervallo nel quale essa può variare)

1.  $R_1 \bowtie_{A=D} R_2$  compresa fra 0 e il minimo fra  $n_1$  e  $n_2$
2.  $R_1 \bowtie_{C=D} R_2$  esattamente  $n_1$
3.  $R_1 \bowtie_{A=F} R_2$  compresa fra 0 e  $n_2$
4.  $R_1 \bowtie_{B=E} R_2$  compresa fra 0 e  $n_1 \times n_2$

#### Domanda 3 (40%)

Considerare una base di dati relativa a studenti ed esami da essi superati:

*Studenti*(Matricola, *Cognome*, *Nome*)

*Esami*(Studente, *Materia*, *Voto*, *Data*)

con vincolo di integrità referenziale fra l'attributo *Studente* di *Esami* e la relazione *Studenti*.

1. Formulare in algebra relazionale l'interrogazione che trova matricola, cognome e nome degli studenti che hanno preso almeno un 30:

$$\pi_{\text{Matricola, Cognome, Nome}}(\text{Studenti} \bowtie_{\text{Matricola}=\text{Studente}} \sigma_{\text{Voto} \geq 30}(\text{Esami}))$$

2. Formulare in SQL l'interrogazione che mostra, per ogni studente, matricola, cognome e media dei voti riportati negli esami superati:

```
SELECT Matricola, Cognome, AVG(Voto) AS Media
FROM Studenti, Esami
WHERE Matricola = Studente
GROUP BY Matricola, Cognome
```

3. Formulare in SQL l'interrogazione che trova gli studenti (mostrare il numero di matricola) che hanno superato almeno due esami dopo il 1/1/2000:

```
SELECT DISTINCT e1.Studente
from Esami e1, Esami e2
where e1. Studente = e2. Studente AND
e1.Data > '2000-01-01' AND e2.Data > '2000-01-01' AND
e1.Materia != e2.Materia
```

oppure

```
SELECT Studente
FROM Esami
WHERE Data > '2000-01-01'
GROUP BY Studente
HAVING COUNT(*) > 1
```

oppure

```
SELECT S.MATRICOLA
FROM STUDENTI S
WHERE 1 < (SELECT COUNT(*)
FROM ESAMI
WHERE MATRICOLA = STUDENTE AND Data > '2000-01-01')
```

4. Scrivere un frammento di programma in SQL immerso in un linguaggio (o pseudolinguaggio) di programmazione che stampi un elenco degli studenti (con matricola, cognome e nome) con, per ciascuno, l'elenco degli esami superati. Gli studenti debbono essere ordinati per matricola e gli

esami per data. Assumere che ciascuno studente abbia superato almeno un esame, ma commentare brevemente come si potrebbe trattare il caso in cui vi siano studenti che non hanno superato esami.

Indichiamo in corsivo istruzioni non specificate esplicitamente

*supponiamo definite nel programma le variabili:*

*vMatricolaPrec, vMatricola, vCognome, vNome, vMateria, vVoto, vData*

```
EXEC SQL declare StudEs cursor for
    SELECT Matricola, Cognome, Nome, Materia, Voto, Data
    FROM Studenti JOIN Esami ON (Matricola = Studente)
    ORDER BY Matricola, Data ;
EXEC SQL open StudEs;
EXEC SQL fetch StudEs
    into :vMatricola, :vCognome, :vNome, :vMateria, :vVoto, :vData ;
if (sqlcode == 0){
    stampa di vMatricola vCognome vNome
    stampa di vMateria vVoto vData
    vMatricolaPrec = vMatricola ;
    EXEC SQL fetch StudEs
        into :vMatricola, :vCognome, :vNome, :vMateria, :vVoto, :vData ;
    while (sqlcode == 0)
    {
        if {vMatricola != vMatricolaPrec}{
            stampa di vMatricola vCognome vNome
            vMatricolaPrec = vMatricola ;
        } ;
        stampa di vMateria vVoto vData
    EXEC SQL fetch StudEs
        into :vMatricola, :vCognome, :vNome, :vMateria, :vVoto, :vData ;
    }
}
$ close cursor StudEs;
```

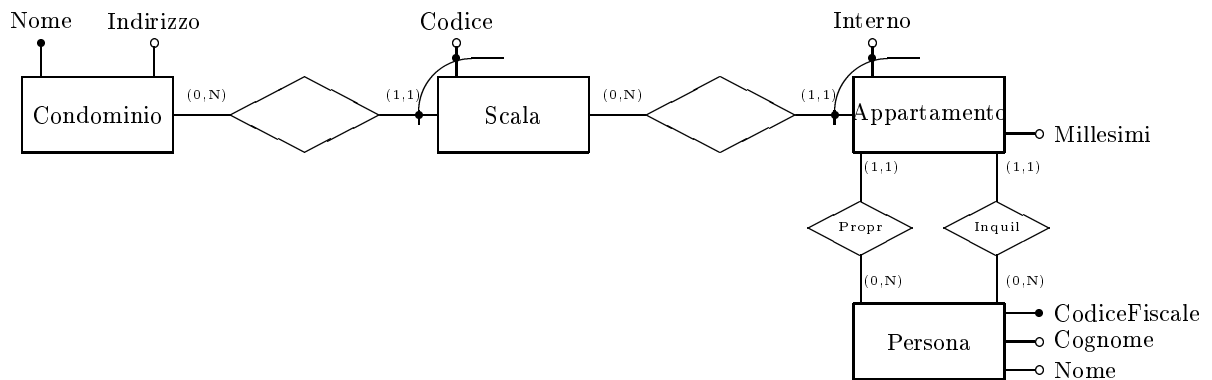
Per includere anche gli studenti che non hanno superato esami si può utilizzare un outer join (sinistro), supponendo che il linguaggio di programmazione sia in grado di riconoscere opportunamente il valore nullo.

**Domanda 4 (30%)**

Definire uno schema Entity-Relationship che descriva i dati di una applicazione relativa all'archivio di un amministratore di condomini, secondo le seguenti specifiche (semplificate rispetto a molte realtà).

- Ogni condominio ha un nome (che lo identifica) e un indirizzo e comprende un insieme di appartamenti raggruppati per "scale".
- Ogni scala è identificata da un codice e dal condominio di appartenenza.
- Ogni appartamento è identificato, nel rispettivo condominio, dalla scala e da un numero (l' "interno"). Ad ogni appartamento sono associati i "millesimi" (che indicano la frazione di condominio che è di proprietà del proprietario dell'appartamento) e un proprietario ed un inquilino (che possono coincidere), ciascuno con codice fiscale, cognome e nome.

Indicare le cardinalità delle relazioni e un identificatore per ciascuna entità.



# Sistemi informativi, primo modulo — 17 novembre 2000

## Compito B

Tempo a disposizione un'ora e trenta minuti. Libri chiusi.

### Soluzioni

**Domanda 1** (15%)

Simile al compito A.

**Domanda 2** (15%)

Simile al compito A.

**Domanda 3** (40%)

Considerare una base di dati relativa a studenti ed esami da essi superati:

*Studenti*(Matricola, *Cognome*, *Nome*)

*Esami*(Studente, Materia, *Voto*, *Data*)

con vincolo di integrità referenziale fra l'attributo *Studente* di *Esami* e la relazione *Studenti*.

1. Formulare in SQL l'interrogazione che trova matricola, cognome e nome degli studenti che hanno preso almeno un 30:

```
SELECT DISTINCT Matricola, Cognome, Nome
FROM Studenti, Esami
WHERE Matricola = Studente
AND Voto = 30
```

oppure

```
SELECT DISTINCT Matricola, Cognome, Nome
FROM Studenti JOIN Esami ON Matricola = Studente
WHERE Voto = 30
```

2. Formulare in SQL l'interrogazione che mostra, per ogni studente, matricola, cognome e numero di esami superati.

```
SELECT Matricola, Cognome, COUNT(*) AS NumeroEsami
FROM Studenti JOIN Esami ON Matricola = Studente
GROUP BY Matricola, Cognome
```

3. Formulare in algebra relazionale l'interrogazione che trova gli studenti (mostrare il numero di matricola) che hanno superato almeno due esami dopo il 1/1/2000.

$$\pi_{Studente}(\sigma_{Data > 1/1/2000 \wedge Data' > 1/1/2000 \wedge Materia <> Materia'}(Esami \bowtie_{Studente=Studente'}(\rho_{Studente', Materia', Data', Voto' \leftarrow Studente, Materia, Data, Voto}(Esami))))$$

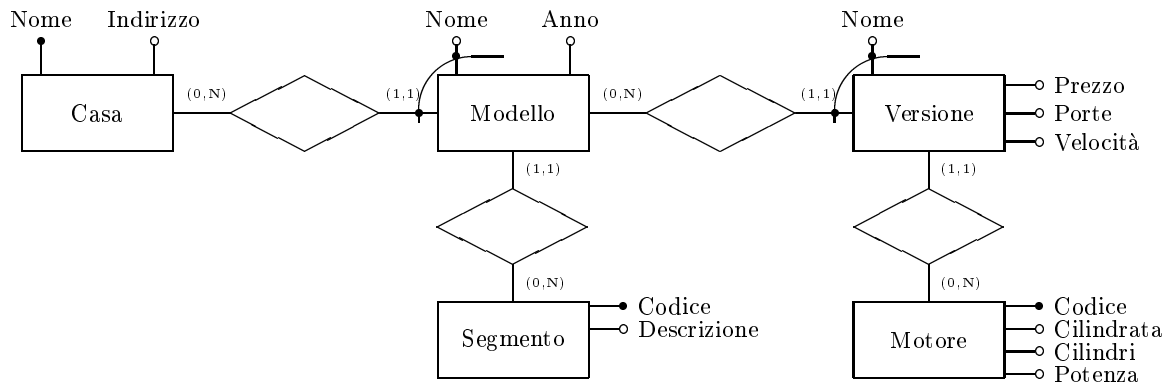
4. Simile a quella del compito A (cambia solo il criterio di ordinamento).

**Domanda 4 (30%)**

Definire uno schema Entity-Relationship che descriva i dati di una applicazione relativa ai listini prezzi di un insieme di case automobilistiche. Sono di interesse:

- Le case produttrici, con nome (identificante) e indirizzo.
- I modelli (ad esempio la Punto o la Golf), con nome, anno di lancio e segmento di mercato (codificato con una lettera e con una breve descrizione: ad esempio, al segmento “A” corrisponde la descrizione “utilitaria”). Il nome identifica univocamente insieme alla casa costruttrice.
- Le versioni dei modelli, identificate attraverso il nome della casa, quello del modello e un nome specifico (ad esempio la Fiat Punto 75S). Per ogni versione sono rilevanti il prezzo, il motore, la cilindrata, la potenza, il numero di porte e la velocità massima. Ogni versione di modello ha uno ed un solo motore.
- I motori (ad esempio il motore Fire 1000), identificati attraverso un codice e con le seguenti proprietà: cilindrata, numero cilindri e potenza. Possono esistere motori (attualmente) non utilizzati in alcun modello.

Indicare le cardinalità delle relazioni e un identificatore per ciascuna entità.



# Sistemi informativi, primo modulo — 17 novembre 2000

## Compito C

Tempo a disposizione un'ora e trenta minuti. Libri chiusi.

### Soluzioni

**Domanda 1** (15%)

Vedi compito A (cambia solo l'ordine delle risposte).

**Domanda 2** (15%)

Vedi compito A (cambia solo l'ordine delle risposte).

**Domanda 3** (40%)

Considerare una base di dati relativa a studenti ed esami da essi superati:

*Studenti*(Matricola, *Cognome*, *Nome*)

*Esami*(Studente, Materia, *Voto*, *Data*)

con vincolo di integrità referenziale fra l'attributo *Studente* di *Esami* e la relazione *Studenti*.

1. Formulare in algebra relazionale l'interrogazione che trova matricola, cognome e nome degli studenti che hanno superato almeno un esame dopo il 1/1/2000.

$$\pi_{Matricola, Cognome, Nome}(Studenti \bowtie_{Matricola=Studente} \sigma_{Data > 1/1/2000}(Esami))$$

2. Vedi compito A.

3. Formulare in SQL l'interrogazione che trova gli studenti (mostrare il numero di matricola) che hanno preso almeno due 30.

```
SELECT DISTINCT e1.Studente
from Esami e1, Esami e2
where e1. Studente = e2. Studente AND
      e1.Voto = 30 AND e2.Voto = 30 AND
      e1.Materia != e2.Materia
```

4. Simile a quella del compito A (cambia solo il criterio di ordinamento).

**Domanda 4** (30%)

Vedi compito A.

# Sistemi informativi, primo modulo — 17 novembre 2000

## Compito D

Tempo a disposizione un'ora e trenta minuti. Libri chiusi.

### Soluzioni

**Domanda 1** (15%)

Simile al compito A.

**Domanda 2** (15%)

Simile al compito A.

**Domanda 3** (40%)

Considerare una base di dati relativa a studenti ed esami da essi superati:

*Studenti*(Matricola, *Cognome*, *Nome*)

*Esami*(Studente, Materia, *Voto*, *Data*)

con vincolo di integrità referenziale fra l'attributo *Studente* di *Esami* e la relazione *Studenti*.

1. Formulare in SQL l'interrogazione che trova matricola, cognome e nome degli studenti che hanno superato almeno un esame dopo il 1/1/2000:

```
SELECT DISTINCT Matricola, Cognome, Nome
FROM Studenti, Esami
WHERE Matricola = Studente AND
      Data > '2000-01-01'
```

2. Vedi compito B.

3. Formulare in algebra relazionale l'interrogazione che trova gli studenti (mostrare il numero di matricola) che hanno preso almeno due 30.

$$\pi_{Studente}(\sigma_{Voto=30 \wedge Voto'=30 \wedge Materia \neq Materia'}(Esami \bowtie_{Studente=Studente'}(\rho_{Studente', Materia', Data', Voto' \leftarrow Studente, Materia, Data, Voto}(Esami))))$$

4. Simile a quella del compito A (cambia solo il criterio di ordinamento).

**Domanda 4** (30%)

Vedi compito B.