

Basi di dati - esame del 10/07/2015 con SOLUZIONI

1

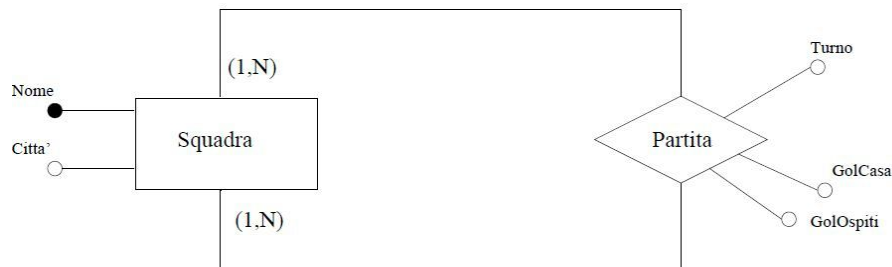
ER) Si consideri il seguente schema relazionale di basi di dati:

Squadra(Nome,Citta)

Partita(SquadraDiCasa,SquadraOspite,GoalCasa,GoalOspiti,Turno)

- a. Riportare (nell'apposito riquadro nel foglio delle risposte) il diagramma E-R relativo allo schema relazionale precedente.

Una possibile soluzione:

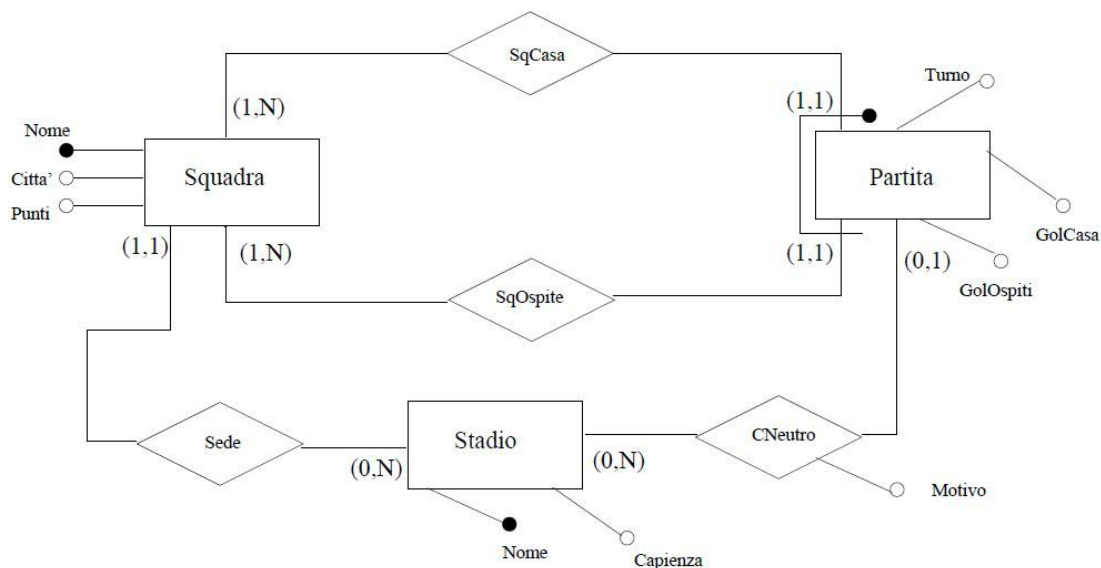


- b. Modificare il diagramma E-R del punto a. aggiungendo le modifiche riportate in seguito (riportare nell'apposito riquadro nel foglio delle risposte il nuovo diagramma E-R). Aggiungere a ciascuna squadra il punteggio del campionato in corso ed i dati relativi al suo stadio (nome, indirizzo e capienza). Uno stadio può appartenere a (in alternativa):

- una sola squadra;
- più squadre;
- a nessuna squadra (ad esempio, uno stadio comunale).

Si consideri anche l'eventualità che una partita si giochi in uno stadio diverso da quello della squadra ("campo neutro"), e in questo caso se ne memorizzi anche il motivo (es. "impraticabilità del campo" o "squalifica").

Una possibile soluzione:



1

Basi di dati - esame del 10/07/2015 con SOLUZIONI

2

TE) Selezionare tutte e solo le risposte corrette (le risposte corrette sono evidenziate in **grassetto**).

a. Nel modello relazionale:

1. Il vincolo di chiave è un vincolo di tupla.
2. **Ogni relazione ha almeno una superchiave.**
3. Ogni relazione ha una sola chiave.
4. Ogni relazione ha una sola superchiave.
5. **Un attributo di una relazione può appartenere a più chiavi.**
6. **Possono esistere attributi che non appartengono a nessuna chiave di una relazione.**
7. **Una superchiave può essere sottoinsieme di un'altra superchiave.**
8. **Può esistere una chiave che coinvolge tutti gli attributi di una relazione.**
9. **Se una chiave coinvolge tutti gli attributi di una relazione, allora è unica.**

b. In algebra relazionale:

1. l'operatore di proiezione preserva i duplicati.
2. In una query di interrogazione SQL, la clausola SELECT corrisponde all'operatore di selezione in algebra relazionale.
3. In una query di interrogazione SQL, la clausola WHERE corrisponde all'operatore di proiezione in algebra relazionale.
4. **Un theta-join con una condizione F tra due relazioni può essere sostituito utilizzando il prodotto cartesiano tra le due relazioni seguito da una selezione con la condizione F.**
5. **Un equi-join è un theta-join in cui la condizione F è un'espressione di uguaglianza tra attributi delle due relazioni coinvolte.**
6. **Gli operatori insiemistici (unione, intersezione e differenza) sono permessi solo se gli operandi coinvolti sono definiti sugli stessi attributi.**
7. L'operatore di ridenominazione cambia nome alle relazioni.

AR) Si consideri lo schema relazionale composto dalle seguenti relazioni:

IMPIEGATO (Matricola, Cognome, Stipendio, Dipartimento)

DIPARTIMENTO (Codice, Nome, Sede, Direttore)

con i seguenti vincoli di integrità referenziale:

- tra Dipartimento della relazione IMPIEGATO e Codice della relazione DIPARTIMENTO.
- tra Direttore della relazione DIPARTIMENTO e Matricola della relazione IMPIEGATO.

Scrivere nel foglio delle risposte le espressioni in algebra relazionale per le seguenti interrogazioni:

- a. Trovare i cognomi degli impiegati che **NON** sono direttori di dipartimento (se un impiegato non direttore di dipartimento ha lo stesso cognome di un direttore di un dipartimento qualsiasi, allora tale cognome **NON** deve comparire).

Una possibile soluzione:

$\pi_{\text{Cognome}}(\text{Impiegato}) - \pi_{\text{Cognome}}(\text{Dipartimento JOIN}_{\text{Direttore=Matricola}} \text{Impiegato})$

- b. Trovare nomi dei dipartimenti in cui lavorano impiegati che guadagnano più di 60000 euro.

Una possibile soluzione:

$\pi_{\text{Nome}}(\sigma_{\text{Stipendio}>60K}(\text{Impiegato JOIN}_{\text{Dipartimento=Codice}} \text{Dipartimento}))$

2

- c. Trovare il cognome degli impiegati che guadagnano più del loro direttore di dipartimento.

Una possibile soluzione:

$$\pi_{\text{Cognome}}(\sigma_{\text{Stipendio} > \text{Stipendio1}}((\text{Impiegato JOIN}_{\text{Dipartimento}=\text{Codice}} \text{Dipartimento}) \\ \text{JOIN}_{\text{Direttore}=\text{Matricola1}} \\ (\rho_{\text{Matricola1}, \text{Cognome1}, \text{Stipendio1}, \text{Dipartimento1} \leftarrow \text{Matricola}, \text{Cognome}, \text{Stipendio}, \text{Dipartimento}} (\text{Impiegato}))))$$

S) Considerando le seguenti tabelle

```
CREATE TABLE elementi(
  codice bigint primary key,
  descrizione varchar(100) not null,
  quantita numeric(8,2) not null);
```

```
CREATE TABLE elementi_della_confezione(
  codice integer primary key,
  tipo_materiale varchar(20) not null,
  riciclabile bit,
  FOREIGN KEY (codice) REFERENCES
  elementi(codice));
```

```
CREATE TABLE indirizzi(
  codice integer primary key,
  indirizzo varchar (50) not null,
  codice_fornitore integer not null,
  FOREIGN KEY (codice_fornitore)
  REFERENCES fornitori(codice));
```

```
CREATE TABLE fornitori(
  codice integer primary key,
  nome varchar(20) not null,
  CF char(16) not null,
  p_iva char (11) not null
);
```

```
CREATE TABLE ingredienti_alimentari(
  codice integer PRIMARY KEY,
  calorie integer,
  FOREIGN KEY (codice) REFERENCES
  elementi(codice));
```

```
CREATE TABLE forniture_correnti(
  codice_fornitore integer not null,
  codice_ingredienti integer not null,
  quantita integer,
  primary key(codice_fornitore,
  codice_ingredienti)
  FOREIGN KEY (codice_fornitore)
  REFERENCES fornitori(codice),
  FOREIGN KEY (codice_ingredienti)
  REFERENCES ingredienti_alimentari(codice));
```

```
CREATE TABLE forniture_passate(
  codice_fornitore integer not null,
  codice_ingredienti integer not null,
  quantita integer,
  PRIMARY KEY(codice_fornitore,
  codice_ingredienti),
  FOREIGN KEY (codice_fornitore)
  REFERENCES fornitori(codice),
  FOREIGN KEY (codice_ingredienti)
  REFERENCES ingredienti_alimentari(codice));
```

Si scelga il codice SQL che implementa le seguenti interrogazioni (le risposte corrette sono evidenziate in **grassetto**).

Basi di dati - esame del 10/07/2015 con SOLUZIONI

4

- a. Trovare la quantità media degli elementi (ingredienti alimentari) forniti da ciascun fornitore in passato, mostrando anche il nome del fornitore. Si escluda dai risultati il fornitore 'pippo' e tutti i fornitori che hanno fornito quantità medie minori di 10.

1 SELECT codice, nome FROM fornitori, forniture_passate WHERE forniture_passate.codice_fornitore=fornitori.codice AND nome <> 'pippo' GROUP BY codice, nome HAVING AVG(forniture_passate.quantita) >=10	2 SELECT codice, nome, AVG(forniture_passate.quantita) AS media FROM fornitori, forniture_passate WHERE forniture_passate.codice_fornitore=fornitori.codice AND nome <> 'pippo' GROUP BY codice, nome HAVING AVG(forniture_passate.quantita) >=10
3 SELECT codice, nome, AVG(forniture_passate.quantita) AS media FROM fornitori, forniture_passate WHERE forniture_passate.codice_fornitore=fornitori.codice AND nome <> 'pippo' GROUP BY codice, nome HAVING AVG(forniture_passate.quantita) <=10	4 SELECT codice, nome, AVG(forniture_passate.quantita) AS media FROM fornitori, forniture_passate WHERE forniture_passate.codice_fornitore=fornitori.codice GROUP BY codice, nome HAVING AVG(forniture_passate.quantita) >=10

- b. Trovare il massimo delle medie degli elementi forniti da ciascun fornitore.

1 CREATE VIEW medie AS SELECT codice, nome AS media FROM fornitori, forniture_passate WHERE forniture_passate.codice_fornitore=fornitori.codice GROUP BY codice, nome SELECT MAX(media) AS massimo FROM medie	2 CREATE VIEW medie AS SELECT codice, nome, AVG(forniture_passate.quantita) AS media FROM fornitori, forniture_passate WHERE forniture_passate.codice_fornitore=fornitori.codice GROUP BY codice, nome SELECT MAX(media) AS massimo FROM medie
3 CREATE VIEW medie AS SELECT codice, nome, AVG(forniture_passate.quantita) AS media FROM fornitori, forniture_passate WHERE forniture_passate.codice_fornitore=fornitori.codice GROUP BY codice, nome SELECT nome, AVG(media) AS massimo FROM fornitori, medie GROUP BY nome, media	4 CREATE VIEW medie AS SELECT codice, nome AS media FROM fornitori, forniture_passate WHERE forniture_passate.codice_fornitore=fornitori.codice GROUP BY codice, nome SELECT nome, MAX(media) AS massimo FROM fornitori, medie GROUP BY nome, media

4

Basi di dati - esame del 10/07/2015 con SOLUZIONI

5

- c. Trovare il nome del fornitore (eventualmente più di uno) che ha fornito un numero di prodotti equivalente al massimo tra le medie.

<p>1</p> <pre>CREATE VIEW medie AS SELECT codice, nome, AVG(forniture_passate.quantita) AS media FROM fornitori, forniture_passate WHERE forniture_passate.codice_fornitore=fornitori.codice GROUP BY codice, nome SELECT nome FROM medie WHERE media=MAX(media)</pre>	<p>2</p> <pre>CREATE VIEW medie AS SELECT codice, nome AS media FROM fornitori, forniture_passate WHERE forniture_passate.codice_fornitore=fornitori.codice GROUP BY codice, nome SELECT nome FROM medie WHERE media=(SELECT MAX(media) FROM medie)</pre>
<p>3</p> <pre>CREATE VIEW medie AS SELECT codice, nome AS media FROM fornitori, forniture_passate WHERE forniture_passate.codice_fornitore=fornitori.codice GROUP BY codice, nome SELECT nome FROM medie WHERE media=MAX(media)</pre>	<p>4</p> <pre>CREATE VIEW medie AS SELECT codice, nome, AVG(forniture_passate.quantita) AS media FROM fornitori, forniture_passate WHERE forniture_passate.codice_fornitore=fornitori.codice GROUP BY codice, nome SELECT nome FROM medie WHERE media=(SELECT MAX(media) FROM medie)</pre>

5

Basi di dati - esame del 10/07/2015 con SOLUZIONI

6

N1) Dato lo schema di relazione $R(A,B,C,D,E,F)$ con le dipendenze funzionali

$AB \rightarrow C$, $A \rightarrow F$, $C \rightarrow D$, $CF \rightarrow AE$, $F \rightarrow BD$, $D \rightarrow E$.

Trovare una copertura ridotta per la relazione R e:

a. Indicare tutte le chiavi (superchiavi minimali) della relazione R

A, CF

b. Segnare con una croce la casella se l'affermazione è vera. N.B. segnare se una dipendenza funzionale viola o meno la forma normale INDIPENDENTEMENTE dalla sua presenza in una copertura ridotta di R .

D.F.	Appartiene alla copertura ridotta	NON viola 3NF	NON viola BCNF
b1. $A \rightarrow B$		X	X
b2. $A \rightarrow C$	X	X	X
b3. $A \rightarrow F$	X	X	X
b4. $A \rightarrow E$		X	X
b5. $C \rightarrow B$			
b6. $C \rightarrow D$	X		
b7. $C \rightarrow E$			
b8. $C \rightarrow F$		X	
b9. $D \rightarrow A$		X	
b10. $D \rightarrow B$			
b11. $D \rightarrow C$		X	
b12. $D \rightarrow E$	X		
b13. $D \rightarrow F$		X	
b14. $CF \rightarrow A$	X	X	X
b15. $CF \rightarrow B$		X	X
b16. $CF \rightarrow D$		X	X
b17. $CF \rightarrow E$		X	X
b18. $F \rightarrow A$		X	
b19. $F \rightarrow B$	X		
b20. $F \rightarrow C$		X	
b21. $F \rightarrow D$	X		

6

N2) Data la relazione $R(A,B,C,D,E,F,G)$ e le dipendenze funzionali

$A \rightarrow B$, $A \rightarrow C$, $A \rightarrow D$, $C \rightarrow D$, $EF \rightarrow A$, $CF \rightarrow E$, $B \rightarrow F$, $B \rightarrow G$.

- a. Le dipendenze funzionali indicate coincidono con una copertura ridotta? Se NO, indicare quali tra le dipendenze funzionali indicate sopra sono ridondanti.

NO, non coincidono. $A \rightarrow D$ è ridondante

- b. Indicare tutte le chiavi (superchiavi minimali) della relazione R.

A, CF, EF, BC, BE

- c. Usando una copertura ridotta per la relazione R, indicare (con una "X" sul foglio delle risposte) le relazioni ottenute dalla decomposizione 3NF:

1. $R_1(A, B, C)$ con $A \rightarrow B$, $A \rightarrow C$
2. $R_2(B, F)$ con $B \rightarrow F$
3. $R_3(B, G)$ con $B \rightarrow G$
4. **$R_4(B, G, F)$ con $B \rightarrow G$, $B \rightarrow F$**
5. **$R_5(C, D)$ con $C \rightarrow D$**
6. $R_6(D, C, F, E)$ con $C \rightarrow D$, $CF \rightarrow E$
7. $R_7(C, F, E, A)$ con $CF \rightarrow E$, $EF \rightarrow A$
8. $R_8(E, C, F)$ con $CF \rightarrow E$
9. $R_9(E, F, A)$ con $EF \rightarrow A$
10. $R_{10}(A, C)$ con $A \rightarrow C$
11. $R_{11}(A, B)$ con $A \rightarrow B$

- d. Indicare con "SI" se tra le precedenti non sono presenti delle relazioni della decomposizione in 3NF (e riportare le relazioni mancanti e le rispettive dipendenze funzionali). Altrimenti scrivere "NO"

SI, manca $R(A,C,B,E,F)$ con $A \rightarrow B$, $A \rightarrow C$, $EF \rightarrow A$, $CF \rightarrow E$

TR1) Data l'esecuzione del seguente codice (il codice non dà errori):

```
CREATE TABLE test1(a1 INT);
CREATE TABLE test2(a2 INT);

DELIMITER |
CREATE TRIGGER testref AFTER INSERT ON test1
FOR EACH ROW
BEGIN
  DECLARE m INT;
  SELECT MIN(a1) FROM test1 INTO m;
  IF m IS NOT Null THEN
    IF NEW.a1 >= 5 THEN INSERT INTO test2(a2) VALUE (m+1);
    ELSEIF NEW.a1 < 4 THEN INSERT INTO test2(a2) VALUE (m+ 2);
    ELSE INSERT INTO test2(a2) VALUE (m + 3);
    END IF;
  END IF;
END IF;
END|
DELIMITER ;

INSERT INTO test1 VALUES (1), (3), (1), (7), (1), (8), (4), (4);
```

Qual'è la sequenza di elementi inseriti nella tabella test2? (segnare nel foglio delle risposte il numero della sola risposta corretta)

1. **3, 3, 3, 2, 3, 2, 4, 4**
2. 3, 3, 3, 2, 3, 2, 3, 4
3. 3, 3, 2, 2, 3, 2, 4, 4
4. 3, 3, 2, 3, 2, 4, 4
5. 3, 3, 3, 2, 3, 2, 4
6. NULL, 3, 3, 2, 3, 2, 4, 4
7. NULL, 3, 3, 3, 2, 2, 2, 4
8. NULL, 3, 2, 2, 3, 2, 4, 4
9. Nessuna delle precedenti (scrivere la risposta corretta nel foglio delle risposte)

TR2) Data l'esecuzione del seguente codice (il codice non dà errori):

```
CREATE TABLE test3(a3 INT);
CREATE TABLE test4(a4 VARCHAR(225));

INSERT INTO test4 VALUES ('BASTONE');

DELIMITER |
CREATE TRIGGER testref2 BEFORE INSERT ON test3
FOR EACH ROW
BEGIN
DECLARE n VARCHAR(255);
SELECT MAX(a4) FROM test4 INTO n;
IF NEW.a3>2 THEN INSERT INTO test4(a4) VALUE (CONCAT(n,'CANE'));
ELSEIF NEW.a3<2 THEN INSERT INTO test4(a4) VALUE (CONCAT(n,'GATTO'));
ELSE INSERT INTO test4(a4) VALUE (CONCAT(n,'TOPO'));
END IF;
END|
DELIMITER ;

INSERT INTO test3 VALUES (3), (1);
```

Indicare il valore del elemento inserito nella tabella test4 all'esecuzione del seguente comando:

```
INSERT INTO test3 VALUE (2);
```

(segnare nel foglio delle risposte il numero della sola risposta corretta)

1. BASTONETOPOCANEGATTO
2. BASTONECANEGATTO
3. BASTONETOPOGATTOCANE
4. BASTONETOPOGATTO
5. BASTONECANETOPOGATTO
6. BASTONETOPOCANE
7. BASTONE
8. BASTONECANEGATTO
9. **Nessuna delle precedenti (scrivere la risposta corretta nel foglio delle risposte)**
BASTONECANEGATTOTOPO

TR3) Data l'esecuzione del seguente codice (il codice non da errori):

```
CREATE TABLE test5 (a5 INT(5) AUTO_INCREMENT PRIMARY KEY, i5 INT(5), s5 VARCHAR(225));  
CREATE TABLE test6 (a6 VARCHAR(225));
```

```
INSERT INTO test5 (i5,s5) VALUES  
    (14,'YES'),(12,'WE'),(10,'REALLY'), (8,'LOVE'), (6,'DATABASE'), (4,'CLASS'),(2,'SO'),(0,'MUCH');
```

```
DELIMITER |  
CREATE TRIGGER testref3 BEFORE UPDATE ON test5  
FOR EACH ROW  
BEGIN  
    DECLARE pippo VARCHAR(255);  
    DECLARE pluto VARCHAR(255);
```

```
SELECT s5 FROM test5 ORDER BY i5 ASC LIMIT 1 INTO pippo;  
SELECT s5 FROM test5 ORDER BY i5 DESC LIMIT 1 INTO pluto;
```

```
IF NEW.i5 >= OLD.i5 THEN INSERT INTO test6(a6) VALUE (pippo);  
ELSE INSERT INTO test6(a6) VALUE (pluto) ;  
END IF;  
END|  
DELIMITER ;
```

```
UPDATE test5 SET i5=a5 WHERE (s5 LIKE '%E_%') OR (s5 LIKE '%_O%');
```

(Hint: la clausola AUTO_INCREMENT comincia con il valore 1 per la prima riga inserita)

Indicare la sequenza di elementi inseriti nella tabella test6 al termine dell'esecuzione del codice riportato sopra (segnare nel foglio delle risposte il numero della sola risposta corretta)

1. YES, WE, LOVE, DATABASE, MUCH
2. YES, WE, LOVE, MUCH, MUCH
3. YES, WE, MUCH, MUCH
4. **YES, WE, WE, MUCH**
5. YES, WE, REALLY, MUCH
6. YES, YES, WE, WE, MUCH
7. WE, WE, MUCH, MUCH
8. LOVE, LOVE, CLASS, MUCH
9. YES, WE, LOVE, DATABASE
10. WE, REALLY, LOVE, DATABASE
11. WE, WE, WE, MUCH
10. Nessuna delle precedenti (scrivere la risposta corretta nel foglio delle risposte)