

Basi di dati (Cod. 861II) 2019-20

Iniziato mercoledì, 22 luglio 2020, 15:33

Tempo dato 40 minuti

Data la tabella T (a, b, c, d) e l'istanza seguente:

<u>a</u>	<u>b</u>	c	d
a1	b2	c1	2
a1	b3	c3	4
a2	b2	c2	5
a1	b1	c1	3
a2	b3	NULL	1

la query:

```
SELECT T.c, LAG(T.b,1) OVER(PARTITION BY T.a ORDER BY T.d) AS N
FROM T
WHERE T.a <> 'a2';
```

Scegli un'alternativa:

- ☒ a. restituisce un result set con due attributi, il secondo dei quali contiene almeno un valore NULL
- ☐ b. nessuna alternativa è corretta
- ☐ c. produce un result set senza valori NULL
- ☐ d. restituisce 2 record
- ☐ e. è errata sintatticamente (non compila)

Data la seguente base di dati di cucina:

```
INGREDIENTE(CodIngr, Nome)
RICETTA(NomeRicetta, Nazionalità)
PROCEDIMENTO(NomeRicetta, Ingrediente)
```

la query:

```
SELECT NomeRicetta
FROM Ricetta NATURAL JOIN Procedimento
WHERE NOT EXISTS( SELECT *
                  FROM Ingrediente);
```

Scegli un'alternativa:

- ☐ a. restituisce il nome delle ricette nel procedimento delle quali manca almeno un ingrediente fra quelli presenti nella tabella INGREDIENTE
- ☒ b. nessuna alternativa è corretta
- ☐ c. restituisce i nomi, eventualmente duplicati, delle ricette per le quali non è ancora stato inserito alcun ingrediente nel database
- ☐ d. non compila
- ☐ e. restituisce i nomi delle ricette per le quali non è ancora stato inserito alcun ingrediente nel database

Siano R(A, B, C) ed S(D, E), le seguenti interrogazioni sono equivalenti

- $\pi_{AB}(R) - \pi_{AB}(R \bowtie_{C=D} S)$
- $\{A:a, B:b \mid \text{not exists } e,c . R(A:a, B:b, C:c) \text{ and } S(D:c, E:e) \}$

Scegli un'alternativa:

- ☒ vero
- ☐ falso

Si consideri lo schema $R(A,B,C,D,E)$ con le dipendenze, $A \rightarrow BC$, $CD \rightarrow E$, $B \rightarrow D$, $E \rightarrow A$, tutte le chiavi di R sono:

Scegli un'alternativa:

- ☒ E
- ☐ Sia A che E
- ☐ CD

Durante la ristrutturazione di uno schema E-R, una generalizzazione senza associazioni fra le entità di livelli diversi

Scegli un'alternativa:

- ☒ a. può sempre essere ristrutturata mantenendo solo l'entità padre
- ☐ b. può essere ristrutturata mantenendo solo l'entità padre solo se la generalizzazione è totale
- ☐ c. non può essere ristrutturata mantenendo solo l'entità padre
- ☐ d. può anche essere mantenuta

Indicare la complessità dell'operazione di verificare se uno schema R soddisfa la 3NF.

Scegli un'alternativa:

- ☐ Costante
- ☐ Polinomiale
- ☒ Esponenziale

Una transazione comprende una o più operazioni di accesso alla base di dati: queste operazioni sono

Scegli un'alternativa:

- ☒ inserzioni, cancellazioni, modifiche e interrogazioni
- ☐ non interrogazioni
- ☐ solo cancellazioni
- ☐ solo inserzioni

In algebra relazionale, il join naturale su relazioni senza attributi in comune

Scegli un'alternativa:

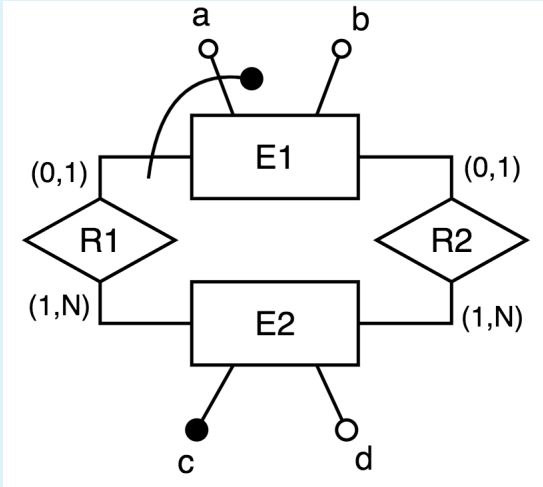
- ☒ è un prodotto cartesiano
- ☐ estende, con valori NULL, le ennuple che verrebbero escluse e le mantiene
- ☐ è un join incompleto
- ☐ nessuna alternativa è corretta
- ☐ è un join vuoto

Considerare lo schema di relazione $R(X)$ e una sua generica istanza r . Dato Y un sottoinsieme non vuoto di X i cui attributi componenti non possono assumere valori NULL, allora, indicando con $| \cdot |$ la cardinalità, l'uguaglianza $|\pi_Y(r)| = |r|$

Scegli un'alternativa:

- ☐ è vera
- ☒ non si può dire a priori se è vera o falsa
- ☐ è falsa

Dato il seguente diagramma E-R



Scegli un'alternativa:

- ☐ a. è ridondante perché si può eliminare R1 (oppure R2) senza perdere informazione
- ☐ b. contiene un ciclo che sicuramente non genera ridondanza
- ☐ c. contiene associazioni ricorsive
- ☐ d. nessuna alternativa è corretta
- ☒ e. è errato

La primitiva *force* viene utilizzata al momento dell'inserzione di un *checkpoint* nel *log* per rendere definitivi i risultati delle transazioni che hanno già fatto *commit* prima di quel momento

Scegli un'alternativa:

- ☐ falso
- ☒ vero

Considerare la tabella S di cui è riportata un'istanza sotto, e il codice della stored procedure `ordina()` riportato sotto.
La generica chiamata `CALL ordina(value)`:

S		
a	b	c
a1	1	c1
a1	2	c1
a1	3	c3
a2	1	(NULL)
a3	1	c2

```
DROP PROCEDURE IF EXISTS ordina;
DELIMITER $$
CREATE PROCEDURE ordina(IN _a CHAR(2))
BEGIN
    DECLARE fine INTEGER DEFAULT 0;
    DECLARE b INTEGER DEFAULT 0;

    DECLARE cur CURSOR FOR
    SELECT S.b FROM S WHERE S.a = _a;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET fine = 1;

    OPEN cur;
    WHILE fine = 0 DO
        FETCH cur INTO b;
        UPDATE S SET S.b = b+1 WHERE S.a = _a;
    END WHILE;
    CLOSE cur;
END $$
```

DELIMITER ;

Scegli un'alternativa:

- ☐ a. aumenta di 1 il valore di S.b solo se value vale 'a1', mentre se value vale 'a2' o 'a3' restituisce un errore
- ☒ b. dipendentemente dal valore di value, può non modificare la tabella
- ☐ c. se value vale 'a4', restituisce un errore
- ☐ d. produce sempre un errore indipendentemente dal valore di value
- ☐ e. non può essere eseguita perché la stored procedure non viene creata a causa di un errore di sintassi

Nella ripresa a caldo nessuna operazione prima del checkpoint può essere disfatta o rifatta

Scegli un'alternativa:

- ☐ vero
- ☒ falso

Adottando il protocollo 2PL stretto

Scegli un'alternativa:

- ☐ a. è possibile, anche se è molto raro, che una transazione abortita lasci il database in uno stato inconsistente
- ☒ b. tutti i lock effettuati da una transazione sono rilasciati dopo il suo commit
- ☐ c. nessuna alternativa è corretta
- ☐ d. si usa un solo lock esclusivo per lettura e scrittura
- ☐ e. è garantita l'assenza di deadlock

Una base di dati ottenuta da un processo di progettazione

Scegli un'alternativa:

- ☐ a. deve avere relazioni tutte in forma normale di Boyce-Codd
- ☐ b. deve avere relazioni tutte in forma normale di Boyce-Codd, o in terza forma normale
- ☒ c. non è necessario che tutte le relazioni siano in forma normale, qualsiasi essa sia
- ☐ d. deve avere relazioni tutte almeno in terza forma normale

Nella ripresa a caldo può accadere che nessuna transazione debba essere ripetuta

Scegli un'alternativa:

- ☐ falso
- ☒ vero

Data la tabella T (a, b, c, d) e l'istanza seguente:

<u>a</u>	<u>b</u>	c	d
a1	b2	c1	4
a1	b3	c3	2
a3	b2	c2	2
a1	b1	c1	7
a2	b3	NULL	12

la query:

```
SELECT T.a, ROW_NUMBER() OVER(PARTITION BY T.a ORDER BY T.d) AS N
FROM T;
```

Scegli un'alternativa:

- ☐ a. numera progressivamente i record di T, restituendo T.a e un numero di riga tanto più grande quanto più è alto il valore di T.d
- ☒ b. produce un result set che contiene duplicati su N
- ☐ c. produce un result set con 3 record, uno per ogni valore di T.a
- ☐ d. nessuna alternativa è corretta

Quando deve essere eseguita un'operazione di una transazione, i dati necessari devono sempre essere trasferiti dalla memoria secondaria al buffer perchè il buffer è vuoto

Scegli un'alternativa:

- ☐ vero
- ☒ falso

Sia data la tabella:
T1(a, b, c, d)

```
la query seguente:
SELECT T1.d
FROM T1
WHERE T1.c > ALL
(
  SELECT T1.c
  FROM T1
);
```

Scegli un'alternativa:

- ☒ a. ha un result set vuoto
- ☐ b. è errata sintatticamente
- ☐ c. può contenere duplicati
- ☐ d. nessuna risposta è corretta
- ☐ e. restituisce tutti i record di T1
- ☐ f. restituisce NULL

Il modello relazionale

Scegli un'alternativa:

- ☒ a. è un modello logico
- ☐ b. è un modello fisico
- ☐ c. è un modello concettuale
- ☐ d. è un modello concettuale, logico e fisico
- ☐ e. nessuna alternativa è corretta

Data la seguente istanza della tabella T, una chiamata alla stored procedure p():

a	b	c	d
a1	b2	1	NULL
a2	b3	3	NULL

a3	b3	1	d2
----	----	---	----

```
DELIMITER $$
CREATE PROCEDURE p()
BEGIN
    SET a = (SELECT SUM(T.c) FROM T WHERE b='b3');
    SELECT a;
END $$
DELIMITER ;
```

- Scegli un'alternativa:
- ☐ a. se eseguita più volte, anche se l'istanza di T resta invariata, stampa valori diversi perché p() non usa parametri d'ingresso
 - ☐ b. stampa a video il numero 1
 - ☐ c. stampa a video il numero 4
 - ☐ d. nessuna alternativa è corretta
 - ☒ e. non può essere eseguita perché la create procedure non compila

Data la tabella S (di cui è mostrata sotto un'istanza), il trigger tr alla sua destra

S		
a	b	c
a1	1	1
a1	4	2
a1	9	3
a2	4	1
a3	2	1

```
CREATE TRIGGER tr
BEFORE INSERT ON S FOR EACH ROW
BEGIN
    SET NEW.c = 1 + (SELECT MAX(c) FROM S WHERE a = NEW.a);
END
```

- Scegli un'alternativa:
- ☐ a. non è mai eseguito perché è errato sintatticamente
 - ☐ b. può essere usato per fare in modo che tutti i record con uguale valore di S.a siano numerati in modo incrementale
 - ☒ c. nessuna alternativa è corretta
 - ☐ d. numera in modo incrementale i record di S all'atto dell'inserimento, assegnando un intero all'attributo c senza che l'utente debba specificare tale attributo nella INSERT
 - ☐ e. è un trigger BEFORE che implementa un vincolo di integrità