

# **PRETEST**

## **(Vaglini, Pistolesi)**



# **INDICE**

---

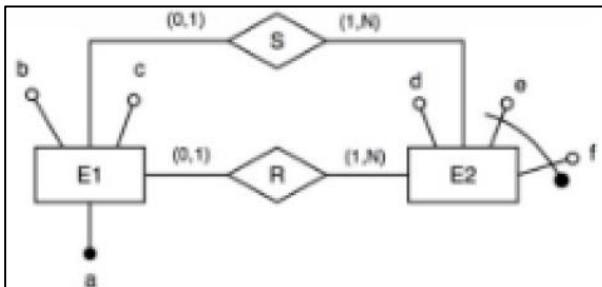
Pretest.....	<b>pag.5</b>
Soluzioni pretest.....	<b>pag.83</b>
Extra (Procedure per esercizi).....	<b>pag.134</b>



# PRETEST

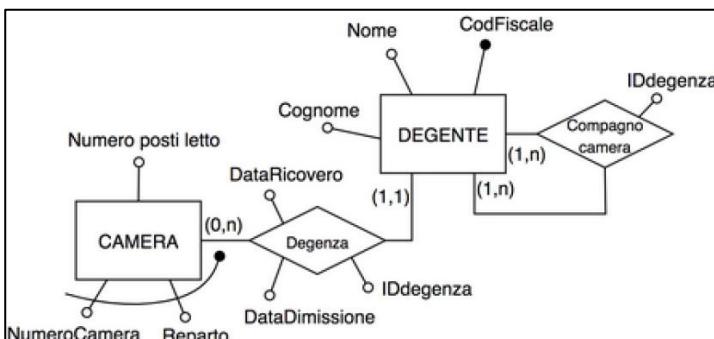
---

1) Dato il diagramma E-R mostrato sotto, scegli un'alternativa corretta:



- a) Contiene ridondanza perché contiene un ciclo
  - b) Nessuna alternativa è corretta
  - c) Anche se è presente un ciclo, in questo caso esso non rappresenta una ridondanza
  - d) Contiene ridondanza perché presenta un ciclo formato da due associazioni identiche, verso le stesse entità, con uguali cardinalità
- 

2) Un degente è un paziente ricoverato in una certa data e dimesso in un'altra data. Ogni volta che un paziente viene ricoverato condivide una stanza, che ha più posti letto, con altri degenenti. Le stanze sono numerate progressivamente all'interno di un reparto. Dato il diagramma E-R seguente, scegli un'alternativa corretta:

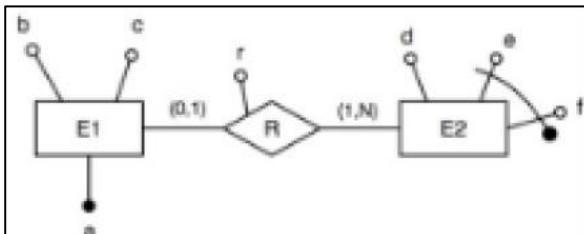


- a) Il diagramma ha la chiave di camera sbagliata perché la cardinalità verso Degenza dovrebbe essere (1,1) non (0,n)
  - b) Nessuna alternativa è corretta
  - c) Il diagramma è errato, tranne la relazione Degenza e le sue cardinalità
  - d) CompagnoCamera nel modello relazionale diventa una tabella con seguente schema: CompagnoCamera(Degente1,Degente2,IDDegenza)
  - e) Il diagramma è corretto perché, traducendolo nel modello logico relazionale, tutti i concetti espressi nella descrizione testuale sono rappresentati, senza ridondanze
  - f) Il diagramma è corretto perché, traducendolo nel modello logico relazionale, tutti i concetti espressi nella descrizione testuale sono rappresentati, ma CompagnoCamera è ridondante
-

# PRETEST

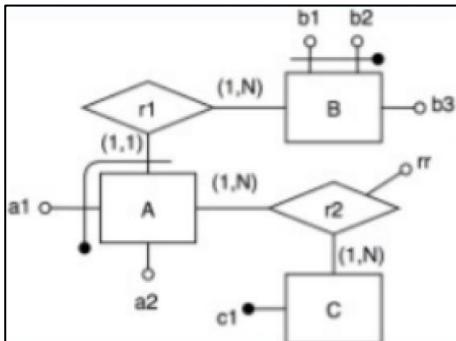
---

3) Nella traduzione del seguente diagramma E-R verso il modello relazionale, scegli un'alternativa:



- a) La relazione R viene sempre eliminata, e l'attributo r finisce nella tabella E1, il cui schema è E1(a,b,c,r); l'entità E2 diventa una tabella con schema E2(e,f,d)
  - b) La relazione R può divenire una tabella
  - c) Si può ottenere il seguente schema di base di dati: E1(a,r,b,c), E2(e,f,d)
  - d) Si può decidere di eliminare R e inserire l'attributo r in E2 perché E1 ha partecipazione opzionale a R
  - e) La relazione R diventa sempre una tabella la cui chiave è l'insieme di attributi (a,e,f)
  - f) Nessuna alternativa è corretta
- 

4) Dato il seguente diagramma E-R, quanti vincoli di integrità referenziale genera nella traduzione verso il modello relazionale? Scegli l'alternativa corretta:

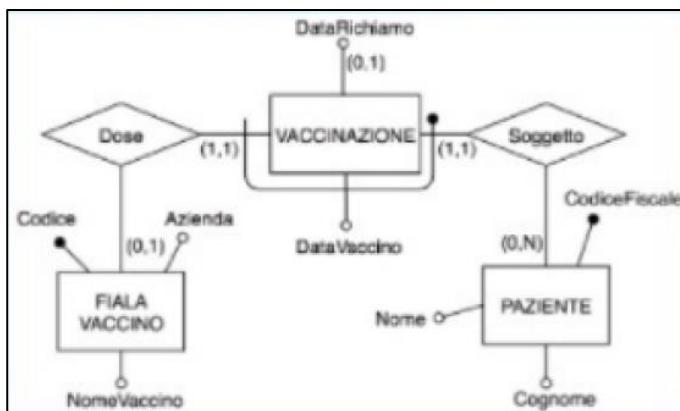


- a) 4
  - b) 2
  - c) 6
  - d) Nessuna alternativa è corretta
  - e) 5
-

# PRETEST

---

5) Un paziente è identificato da un codice fiscale, e ha un nome e cognome. Una fiala di vaccino ha un codice che la identifica, ha un nome e un'azienda farmaceutica che la produce. Un paziente effettua una vaccinazione quando gli viene iniettata una dose di vaccino (fiala), in una certa data. Una vaccinazione può prevedere una dose di richiamo, dopo alcuni giorni. Considerato il seguente diagramma E-R per rappresentare la realtà sopra descritta si può dire che:



- a) Nessuna alternativa è corretta
  - b) L'entità VACCINAZIONE può essere trasformata in un'associazione many-to-many fra PAZIENTE e FIALA VACCINO, con gli attributi DataVaccino e DataRichiamo
  - c) Lo schema è corretto ma, a causa delle cardinalità VACCINAZIONE vincola a vaccinare un solo paziente in una DataVaccino
  - d) Il diagramma è corretto, ma DataRichiamo è un attributo opzionale, e quindi da ristrutturare perché non traducibile nel modello relazionale
  - e) Il diagramma è corretto
- 

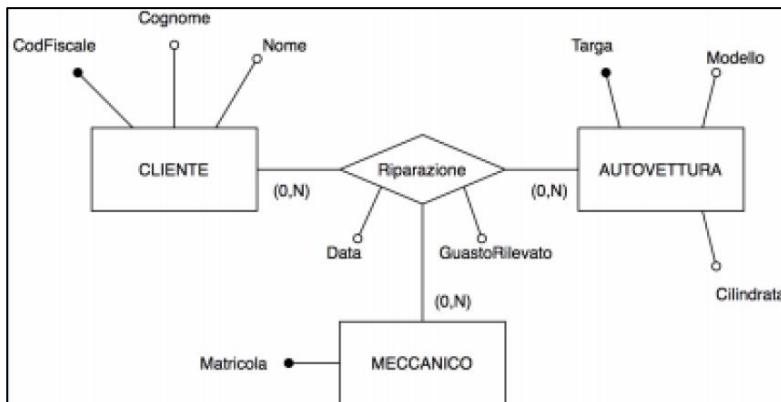
6) In uno schema E-R, eventuali attributi di una associazione **A** fra due entità **E1** ed **E2** fanno parte della chiave della tabella corrispondente ad **A** nello schema logico. Scegli l'alternativa corretta:

- a) Nessuna alternativa è corretta
  - b) Sì, ma la cardinalità minima di E1 o quella minima di E2 deve essere 1
  - c) Dipende dalle cardinalità con cui E1 ed E2 partecipano all'associazione A
  - d) Sempre
  - e) Mai
-

# PRETEST

---

7) In un'officina si effettuano riparazioni di autovetture che hanno una targa, un modello e una cilindrata. I clienti hanno un codice fiscale, un nome e un cognome. Una riparazione è effettuata in una certa data e rileva un certo guasto. In una riparazione, un solo tecnico ripara una sola autovettura, ma ovviamente un utente può portare l'autovettura a riparare più volte. Il diagramma seguente è: (scegli un'alternativa)



- a) È errato perché le cardinalità con cui tutte le entità partecipano alla relazione Riparazione dovrebbero essere (1,1), visto che una riparazione è caratterizzata da una autovettura, un meccanico e un cliente
- b) Nessuna alternativa è corretta
- c) È errato perché se in una riparazione lavora un solo meccanico su una sola autovettura, le cardinalità di AUTOVETTURA e MECCANICO dovrebbero essere entrambe (1,1) e non (0,N)
- d) È giusto

---

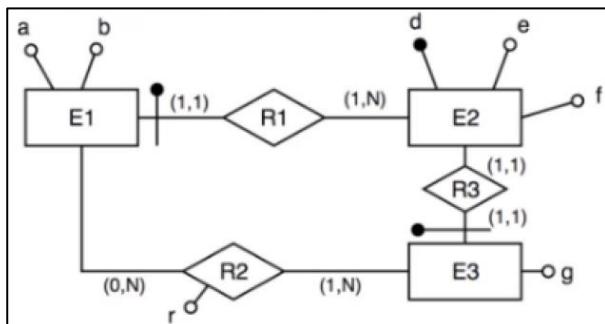
8) Durante la ristrutturazione di uno schema E-R, una generalizzazione senza associazioni fra le entità di livelli diversi:

- a) Può essere sempre 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

# PRETEST

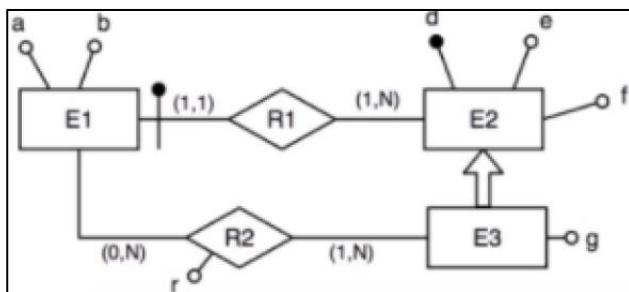
---

9) Dato il seguente diagramma E-R, una sua possibile traduzione nel modello logico relazionale è:



- a) E1(d,a,b), E2(d<sub>1</sub>,d<sub>2</sub>,e,f), E3(d,g,r), R2(d<sub>1</sub>,d<sub>2</sub>,r)
  - b) E1(d,a,b), E2(d,e,f), E3(d,g), R2(d<sub>1</sub>,d<sub>2</sub>,r)
  - c) Nessuna alternativa è corretta
  - d) E1(d,a,b), E2(d,e,f), E3(d,g), R2(d<sub>1</sub>,d<sub>2</sub>,r)
  - e) E1(d,a,b), E2(d,e,f), E3(d,g,r), R1(d<sub>1</sub>,d<sub>2</sub>), R2(d<sub>1</sub>,d<sub>2</sub>,r)
- 

10) Dato il seguente diagramma E-R, scegli un'alternativa



- a) Se la generalizzazione presente viene ristrutturata sostituendola con un'associazione fra E2 ed E3, si introducono 4 vincoli di integrità referenziale
  - b) Contiene ridondanza
  - c) Nessuna alternativa è corretta
  - d) Contiene una generalizzazione totale E2 ed E3
  - e) Se la generalizzazione presente viene ristrutturata sostituendola con un'associazione fra E2 ed E3, tale associazione ha obbligatoriamente cardinalità (1,1) da ambo le parti
- 

11) Si consideri il seguente insieme di dipendenze **AB**→**C**, **AD**→**EF**, **B**→**A**, **A**→**D**, **B**→**E**, quale degli insiemi seguenti è ad esso equivalente? Scegli un'alternativa:

- a) **B**→**A**, **A**→**DEF**
  - b) **B**→**AC**, **A**→**DEF**, **B**→**E**
  - c) Nessuno
  - d) **B**→**C**, **A**→**DF**, **B**→**A**, **B**→**E**
-

# PRETEST

---

12) Per verificare che due insiemi di dipendenza **F** e **G**, sono equivalenti si può utilizzare il concetto di chiusura transitiva degli insiemi, cioè che  $F^+ = G^+$ ; un metodo alternativo è quello della chiusura di insiemi di attributi: che vantaggio c'è ad usarlo? Scegli un'alternativa:

- a) Non c'è differenza tra i due metodi in termini di complessità di calcolo
  - b) Nessuna alternativa è corretta
  - c) Per ogni dipendenza  $X \rightarrow Y$  in F, si calcola  $Y^{+G}$ , e viceversa; quindi, questo metodo è polinomiale
  - d) Per ogni dipendenza  $X \rightarrow Y$  in F, si calcola  $Y^{+G}$ , e viceversa; quindi, questo metodo è lineare
- 

13) Si consideri il seguente insieme di dipendenze **A**→**BCD**, **CD**→**E**, **B**→**CD**, quale degli insiemi seguenti è ad esso equivalente? Scegli un'alternativa:

- a) **A**→**B**, **CD**→**E**, **B**→**CD**
  - b) **A**→**BC**, **CD**→**E**, **B**→**D**
  - c) Nessuna alternativa è corretta
  - d) **A**→**BC**, **CD**→**E**, **B**→**CD**
- 

14) Si consideri lo schema **R(A,B,C,D,E)** con le dipendenze, **A**→**BC**, **CD**→**E**, **B**→**D**, **E**→**A**, tutte le chiavi di **R** sono:

- a) E
  - b) Sia A che E
  - c) CD
- 

15) Si consideri il seguente insieme di dipendenze **A**→**BCD**, **CD**→**E**, **B**→**D**, **A**→**E**, quale degli insiemi seguenti è l'insieme minimale ad esso equivalente? Scegli un'alternativa:

- a) **A**→**BCD**, **CD**→**E**, **B**→**D**
  - b) **A**→**BC**, **CD**→**E**, **A**→**E**
  - c) **A**→**BC**, **CD**→**E**, **B**→**D**
  - d) Nessuno è minimale
- 

16) Si consideri il seguente insieme di dipendenze **A**→**BCDE**, **CD**→**E**, **B**→**D**, quale degli insiemi seguenti è l'insieme minimale ad esso equivalente? Scegli un'alternativa:

- a) Nessuna alternativa è corretta
  - b) L'insieme è già minimale
  - c) **A**→**BCE**, **CD**→**E**, **B**→**D**
  - d) **A**→**BC**, **CD**→**E**, **B**→**D**
-

# PRETEST

---

17) Data l'istanza della tabella **T1(a,b,c,d)** mostrata sotto, la query alla sua destra:

T1			
a	b	c	d
a1	b1	c1	7
a1	b2	c1	4
a1	b3	c3	2
a2	b3	(NULL)	12

```
SELECT T1.b, COUNT(*)
FROM T1
WHERE T1.d < 10
GROUP BY T1.a
HAVING COUNT(DISTINCT T1.c) = 2;
```

- a) Può proiettare T1.b perché il suo valore, essendo nella chiave, dipende da T1.a
  - b) Conta i record che hanno due valori diversi sull'attributo T1.c
  - c) Nessuna alternativa è corretta
  - d) Per ogni gruppo di record aventi uno stesso valore sull'attributo T1.a, restituisce il corrispondente valore T1.b e conta, nel gruppo, i record che hanno due valori diversi di T1.c
- 

18) Data l'istanza della tabella **T** mostrata sotto, la query alla sua destra:

T			
a	b	c	d
a1	b1	9	2021-01-02
a1	b2	1	2020-12-20
a1	b5	0	2021-01-31
a2	b1	3	2021-02-01
a2	b3	2	2020-11-18

```
SELECT T.* , T2.d
FROM T
WHERE EXISTS (SELECT *
               FROM T T2
               WHERE T2.d < T.d);
```

- a) Per ogni record t di T per cui esiste un record di T con data antecedente, restituisce tutti gli attributi di t e la data del primo record con data antecedente trovato
- b) Restituisce 4 record perché per il record con T.d più remoto non esiste un record con T.d antecedente
- c) Non compila
- d) Nessuna alternativa è corretta

# PRETEST

---

19) Data l'istanza della tabella **T** mostrata sotto, la query alla sua destra:

T			
a	b	c	d
a1	b1	c2	3
a1	b3	c4	9
a1	b7	c1	2
a2	b1	(NULL)	2
a3	b2	c2	5

```
SELECT T.a, SUM(T.d)
FROM T
WHERE AVG(T.d) > 2
GROUP BY T.a;
```

- a) Per ogni valore di T.a, restituisce T.a e la somma dei valori di T.d per i soli record che hanno quel valore medio di T.d superiore a 2
  - b) Nessuna alternativa è corretta
  - c) Contiene nel SELECT attributi che non possono essere proiettati
  - d) Elimina i record di T che hanno valore medio su T.d minore o uguale a 2 e poi raggruppa su T.a, restituendo T.a e la somma dei valori su T.d
- 

20) Sia data la tabella **T1(a,b,c,d)** e la seguente istanza **T1**, la query che segue :

a	b	c	d
a1	b2	c1	d1
a3	b3	c3	d2
a2	b2	c1	d1
a1	b1	c2	d3
a1	b3	c1	NULL

```
SELECT T1.a, COUNT(*)
FROM T1
WHERE T1.d IS NULL
GROUP BY T1.a;
```

- a) Conta quanti record con T1.d pari a NULL ci sono in ogni gruppo di record entro il quale tutti i record assumono un dato valore sull'attributo T1.a e un qualsiasi valore sui restanti attributi
- b) Conta i record che hanno valore NULL sull'attributo T1.d
- c) Nessuna alternativa è corretta
- d) Restituisce il seguente record: (a1,NULL)
- e) È errata
- f) Può contenere duplicati

# PRETEST

---

21) Sia data la tabella **T1(a,b,c,d)**. Considerando l'attributo ‘c’, il result set della query seguente:

```
SELECT COUNT(DISTINCT T1.d)
FROM T1
WHERE T1.c > ALL
(
  SELECT T1.c
  FROM T1
);
```

- a) Restituisce 0
  - b) Potrebbe contenere duplicati dipendentemente dall'istanza di T1
  - c) Restituisce NULL
  - d) Conta i record diversi di T1
  - e) Non viene prodotto dato che la query è errata perché la subquery non usa un alias per T1
- 

22) Data la tabella **T** nello stato mostrato sotto a sinistra, l'esecuzione della query alla sua destra:

T			
a	b	c	d
a1	b1	9	2021-01-02
a1	b2	1	2020-12-20
a1	b5	0	2021-01-31
a2	b1	3	2021-02-01
a2	b3	2	2020-11-18

```
SELECT T1.*, T2.a
FROM T T1 LEFT OUTER JOIN T T2
ON T1.a <> T2.a;
```

- a) Produce 10 record perché a ciascun record t1 di T associa il primo record t2 di T che fa join tale che t1.a <> t2.a
  - b) Produce almeno un record con T2.a pari a NULL
  - c) Produce i 5 record di T, con i loro valori per gli attributi di T, e sempre NULL come valore di T.a
  - d) Produce 12 record, nessuno dei quali con valore di T2.a pari a NULL
  - e) Nessuna alternativa è corretta
-

# PRETEST

---

23) 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);
```

- 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 nei database
  - d) Non compila
  - e) Restituisce i nomi delle ricette per le quali non è ancora stato inserito alcun ingrediente nei database
- 

24) 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
);
```

- 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
-

# PRETEST

---

25) Nell'istanza H di sotto a sinistra, la query alla sua destra:

H				
a	b	c	d	e
a1	b1	c1	7	2021-01-02
a1	b2	c3	2	2020-12-20
a1	b3	c3	1	NULL
a1	b5	NULL	2	2021-01-31
a2	b1	c1	4	2021-02-01
a2	b3	c2	5	2020-11-18
NULL	NULL	NULL	NULL	NULL

```
SELECT a, b, RANK() OVER (PARTITION BY a ORDER BY d) AS r  
FROM H  
WHERE a <> 'a2';
```

- a) Nessuna alternativa è corretta
  - b) Restituisce un result set con tre valori diversi di 'r'
  - c) È errata sintatticamente
  - d) Restituisce un result set senza duplicati su 'r' perché il rank non è dense
  - e) Per ogni valore di 'a' della tabella H, restituisce un rank con gap tanto più piccolo quanto più basso il valore di 'c'
- 

26) Nell'istanza H(a,b,c,d,e) di sotto a sinistra, la query alla sua destra:

H				
a	b	c	d	e
a1	b1	c1	7	2021-01-02
a1	b2	c3	2	2020-12-20
a1	b3	c3	1	NULL
a1	b5	NULL	2	2021-01-31
a2	b1	c1	4	2021-02-01
a2	b3	c2	5	2020-11-18
NULL	NULL	NULL	NULL	NULL

```
SELECT a, b, c, LEAD(c,1) OVER(PARTITION BY a ORDER BY d)  
FROM H  
WHERE c IS NOT NULL OR e IS NOT NULL;
```

- a) Restituisce un result set in cui la colonna 'c' contiene tutti i valori dell'attributo 'c' presenti in H
  - b) Restituisce c=NULL solo per un record, l'ultimo record del result set, perché la funzione lead non può anticipare il valore di 'c' del record successivo in quanto tale valore non esiste
  - c) È errata sintatticamente
  - d) Restituisce 3 record
  - e) Nessuna alternativa è corretta
  - f) Restituisce 4 record
-

# PRETEST

---

27) Nell'istanza **H(a,b,c,d,e)** di sotto a sinistra, la query alla sua destra:

H				
a	b	c	d	e
a1	b1	c1	7	2021-01-02
a1	b2	c3	2	2020-12-20
a1	b3	c3	1	NULL
a1	b5	NULL	2	2021-01-31
a2	b1	c1	4	2021-02-01
a2	b3	c2	5	2020-11-18

```
SELECT H1.a, H1.c
FROM H H1
WHERE H1.a = (SELECT H2.a
               FROM H H2
               WHERE H1.e < H2.e
                     AND H1.a = H2.a
                     AND H1.b = H2.b);
```

- a) Non compila perché la subquery è noncorrelated
  - b) Non compila perché la subquery non è scalare
  - c) Compila solo se, per legare la subquery all'outer query, si sostituisce '=' con 'IN'
  - d) Nessuna alternativa è corretta
- 

28) Siano date le tabelle **T1(a,b,c,d)** e **T2(e,f,g,h,i)**, la query seguente:

```
SELECT DISTINCT(T1.b)
FROM T1
WHERE T1.c IN
(
  SELECT T2.g
  FROM T2
  WHERE T2.i <> T1.d
);
```

- a) Restituisce un numero
  - b) Nessuna risposta è corretta
  - c) È errata sintatticamente
  - d) Deve avere un result set scalare nella subquery
  - e) Calcola il result set della subquery per ogni record di T1 e controlla che T1.c sia fra i valori T2.g nel resul set della subquery
- 

29) Data la tabella **H(a,b,c,d,e)** di sotto a sinistra, la query alla sua destra:

H				
a	b	c	d	e
a1	b4	c2	5	2021-01-20
a1	b2	c1	2	2020-12-02
a1	b3	c2	1	NULL
a1	b5	NULL	2	2020-01-03
a1	b1	c1	1	2020-03-01
a2	b3	c1	5	2021-05-28

```
SELECT H1.a, H1.b
FROM H H1
WHERE NOT EXISTS (
  SELECT *
  FROM H H2
  WHERE H2.a <> H1.a
        AND H2.b <> H2.c
);
```

- a) Restituisce tutti i valori della chiave di H
  - b) Restituisce tutti i valori della chiave di H, tranne quelli con valore NULL sull'attributo 'c'
  - c) Restituisce un result set vuoto
  - d) Nessuna alternativa è corretta
  - e) È errata sintatticamente, non compila
-

# PRETEST

---

30) Data la tabella **H(a,b,c,d,e)** di sotto a sinistra, la query alla sua destra:

H				
a	b	c	d	e
a1	b4	c2	5	2021-01-20
a1	b2	c1	2	2020-12-02
a1	b3	c2	1	(NULL)
a1	b5	(NULL)	2	2020-01-03
a1	b1	c1	1	2020-03-01
a2	b3	c1	5	2021-05-28

```
SELECT d,a,b  
FROM H  
WHERE e > '2020-01-01'  
GROUP BY d  
HAVING MAX(d) = 5;
```

- a) Nessuna alternativa è corretta
- b) Restituisce un record per ogni record di H
- c) Restituisce un result set vuoto
- d) Restituisce un record
- e) Proietta attributi che non sono nel predicato GROUP BY, ma questo non è un errore perché tali attributi sono la chiave primaria di H: per ogni valore di 'd' esiste una sola coppia di valori di 'a' e 'b'

31) Data la tabella **P** sotto a sinistra, il cui schema è **P(a,b,c,d,e)**, la query alla sua destra:

P				
a	b	c	d	e
a1	b1	c1	2	2021-01-03
a1	b2	NULL	2	2020-05-30
a1	b3	c3	4	2021-03-31
a2	b2	c2	5	NULL
a2	b3	NULL	1	2020-07-29
a3	b2	c2	1	2021-04-17

```
SELECT d  
FROM P  
WHERE a<>'a1' AND b='b2'  
GROUP BY d  
HAVING COUNT(DISTINCT a) > 1;
```

- a) Nessuna alternativa è corretta
- b) Restituisce un solo record con valore di 'd' pari a 1
- c) È errata sintatticamente
- d) Restituisce NULL
- e) Restituisce l'intera colonna 'd' perché tutti i gruppi, per definizione di raggruppamento, hanno un numero di record maggiore di 1

# PRETEST

---

32) Data l'istanza di T mostrata sotto a sinistra, la query alla sua destra:

T			
a	b	c	d
a1	b1	c2	3
a1	b3	c4	9
a1	b7	c1	2
a2	b1	(NULL)	2
a3	b2	c2	5

```
CREATE EVENT E
ON SCHEDULE AT '2021-01-15 23:59:59'
DO
    UPDATE T
    SET T.d = (SELECT SUM(T2.d)
                FROM T AS T2
                WHERE T2.a = T.a
              );
```

- a) È errato sintatticamente
  - b) Fa sì che T.d sia calcolato ogni giorno in automatico a partire dalle 23:59:59 del 15-01-2021
  - c) Vieta all'utente di modificare manualmente i valori di T.d
  - d) Nessuna alternativa è corretta
  - e) Crea un event che, per ogni record t di T, mantiene aggiornato l'attributo T.d impostandolo alla somma dei valori di T.d di tutti i record di T, compreso t
- 

33) Data l'istanza di T mostrata sotto a sinistra, la query che la segue:

a	b	c	d
a1	b2	c1	NULL
a2	b3	c1	NULL

```
DELIMITER $$ 
CREATE PROCEDURE p(OUT ris_ INT)
BEGIN
    SET ris_ = (SELECT DISTINCT T.c FROM T);
END $$ 
DELIMITER ;
CALL p(@a);
SELECT @a;
```

- a) Stampa a video NULL
  - b) Stampa a video 1
  - c) Stampa a video 2
  - d) Nessuna alternativa è corretta
  - e) Non si può sapere cosa stampa perché la variabile @ non è inizializzata
-

# PRETEST

34) 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 ;
```

- 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

# PRETEST

---

35) 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 ;
```

- a) Se eseguita più volte, anche se l'istanza di T resta invariata, stampa valori diversi perché p() non usa parametri di 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
- 

36) L'esecuzione del seguente codice

```
DELIMITER $$  
CREATE PROCEDURE p(IN _a INT, IN _b INT)  
BEGIN  
    SET @r = '';  
    REPEAT SET @r = CONCAT(@r,_a+1);  
    UNTIL _a < _b  
    END REPEAT;  
END $$  
DELIMITER ;  
  
CALL p(1,5);  
SELECT @r;
```

- a) Nessuna alternativa è corretta
  - b) Stampa una stringa costruita concatenando le cifre da 1 a 5
  - c) Stampa NULL perché il valore di @r impostato nella stored procedure non è visibile dall'esterno
  - d) La stored procedure va in loop infinito
  - e) Stampa la stringa '1'
-

# PRETEST

---

37) Data la tabella **T** la stored function che segue:

T			
a	b	c	d
a1	b1	9	2021-01-02
a1	b2	1	2020-12-20
a1	b5	0	2021-01-31
a2	b1	3	2021-02-01
a2	b3	2	2020-11-18

```
DELIMITER $$  
CREATE FUNCTION f(_c INT)  
RETURNS INT  
BEGIN  
    SET @res = (SELECT SUM(T.c)  
                FROM T  
                WHERE T.c <> _c  
                GROUP BY T.a);  
    RETURN @res;  
END $$  
DELIMITER ;
```

- a) È deterministic
  - b) Nessuna alternativa è corretta
  - c) È non deterministic
  - d) Restituisce un errore a tempo di esecuzione
- 

38) 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
```

- 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
-

# PRETEST

---

39) Data la tabella **T(a,b,c,d)** e l'istanza seguente, la query alla sua destra:

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

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

- a) È errata sintatticamente (non compila)
  - b) Restituisce un result set con due attributi, in nessuno dei quali compaiono valori 'b2' e 'c2'
  - c) Restituisce un solo record
  - d) Nessuna alternativa è corretta
  - e) Produce un result set senza valori NULL
- 

40) Data la tabella **T(a,b,c,d)** e l'istanza seguente, la query alla sua destra:

a	b	c	d
a1	b2	NULL	2
a1	b4	c3	4
a2	b2	c2	5
a1	b1	c1	2
a2	b3	NULL	1

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

- a) Produce un result set in cui nessun record ha rank superiore a 2
  - b) Produce un result set che effettua un rank di tutti i record di T, in cui la posizione in classifica di un record è tanto più alta quanto più è basso il valore dell'attributo d
  - c) È errata sintatticamente (non compila)
  - d) Nessuna alternativa è corretta
  - e) Produce un result set che non contiene ex-aequo (pari merito)
- 

41) Data la tabella **T** e l'istanza seguente, la query alla sua destra:

T			
a	b	c	d
a1	b1	9	2021-01-02
a1	b2	1	2020-12-20
a1	b5	0	2021-01-31
a2	b1	3	2021-02-01
a2	b3	2	2020-11-18

```
SELECT T.* , LAST_VALUE(T.c)
OVER(PARTITION BY T.a ORDER BY T.d) AS FV
FROM T;
```

- a) Restituisce 5 record tutti con FV diverso
  - b) Produce un record per ogni valore di T.a
  - c) Per ogni valore di T.a crea una partition con tutti i valori di T con quel valore T.a e il valore di T.c associato al record con T.d più recente fra quelli con quel valore di T.a
  - d) Nessuna alternativa è corretta
  - e) Restituisce valori di FV sempre uguali a 3 o 9
-

# PRETEST

---

42) Data la tabella **T** e l'istanza seguente, la query alla sua destra:

T				
a	b	c	d	
a1	b1	9	2021-01-02	
a1	b2	1	2020-12-20	
a1	b5	0	2021-01-31	
a2	b1	3	2021-02-01	
a2	b3	2	2020-11-18	

```

SELECT T.a, T.b,
       AVG(T.c) OVER(PARTITION BY T.a
                      ORDER BY T.d
                      RANGE BETWEEN UNBOUNDED PRECEDING
                      AND UNBOUNDED FOLLOWING) AS M
FROM T;
    
```

- a) Restituisce un unico valore di M per tutti i record di ciascuna partition
  - b) Per ciascun record di una partition, restituisce un unico valore di M potenzialmente diverso dagli altri record per la stessa partition
  - c) Non compila
  - d) Nessuna alternativa è corretta
- 

43) Data l'istanza di **P** sotto a sinistra, la query alla sua destra:

P				
a	b	c	d	e
a1	b1	c1	2	2021-01-03
a1	b2	NULL	2	2020-05-30
a1	b3	c3	4	2021-03-31
a2	b2	c2	5	NULL
a2	b3	NULL	1	2020-07-29
a3	b2	c2	1	2021-04-17

```

SELECT a, b, AVG(d) OVER(PARTITION BY a ORDER BY d)
FROM P
WHERE c <> 'c1';
    
```

- a) Restituisce lo stesso risultato se si toglie l'ordinamento della partition
  - b) Nessuna alternativa è corretta
  - c) È sintatticamente errata
  - d) Restituisce un unico record
  - e) Non restituisce lo stesso risultato se si toglie l'ordinamento della partition perché la query lavorerebbe sul default frame
- 

44) Data l'istanza di **P** sotto a sinistra, le due query (1 e 2) alla sua destra:

P				
a	b	c	d	e
a1	b1	c1	2	2021-01-03
a1	b2	NULL	2	2020-05-30
a1	b3	c3	4	2021-03-31
a2	b2	c2	5	NULL
a2	b3	NULL	1	2020-07-29
a3	b2	c2	1	2021-04-17

```

-- query 1
SELECT a, b, SUM(d) OVER w
FROM P
WHERE d < 5
WINDOW w AS
(PARTITION BY a ORDER BY e
 ROWS BETWEEN 1 PRECEDING AND CURRENT ROW);

-- query 2
SELECT a, b, d + LAG(d,1) OVER(PARTITION BY a ORDER BY e)
FROM P
WHERE d < 5;
    
```

- a) Nessuna alternativa è corretta
  - b) Non restituiscono lo stesso risultato, ma potrebbero restituirlo se l'istanza di P fosse diversa
  - c) Restituiscono sempre risultati diversi
  - d) Restituiscono sempre lo stesso risultato perché sono due modi alternativi per fare la stessa cosa
-

# PRETEST

---

45) Data la tabella **T(a,b,c,d)** e l'istanza seguente, la query alla sua destra:

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

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

- 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)
- 

46) Data la tabella **T(a,b,c,d)** e l'istanza seguente, la query alla sua destra:

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

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

- 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
- 

47) Data la tabella **T(a,b,c,d)** e l'istanza seguente, la query alla sua destra:

a	b	c	d
a1	b2	NULL	2
a1	b4	c3	4
a2	b2	c2	5
a1	b1	c1	3
a2	b3	NULL	1

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

- a) Nessuna alternativa è corretta
  - b) Produce un result set che contiene ex-aequo (pari merito)
  - c) Effettua un rank di tutti i record della tabella T in base al valore dell'attributo 'd'
  - d) Produce un result set che non contiene ex-aequo (pari merito)
  - e) È errata sintatticamente (non compila)
-

# PRETEST

---

48) Data l'istanza di H(a,b,c,d,e) sotto a sinistra, le due query alla sua destra:

H				
a	b	c	d	e
a1	b4	c2	5	2021-01-20
a1	b2	c1	2	2020-12-02
a1	b3	c2	1	(NULL)
a1	b5	(NULL)	2	2020-01-03
a1	b1	c1	1	2020-03-01
a2	b3	c1	5	2021-05-28

```
-- query 1  
SELECT a, LAST_VALUE(d) OVER(PARTITION BY a) AS L,  
      FIRST_VALUE(d) OVER(PARTITION BY a) AS F  
FROM H  
WHERE c <> 'c2';  
  
-- query 2  
SELECT a, d AS L, d AS F  
FROM H  
WHERE c <> 'c2';
```

- a) Nessuna alternativa è corretta
- b) Restituirebbero lo stesso risultato se la prima query partizionasse anche per 'b', in entrambi i 'partition by'
- c) Restituiscono lo stesso risultato
- d) Non restituiscono lo stesso record solo per i record non appartenenti a partizioni composte da più record; per gli altri record, il result set è identico
- e) Restituiscono lo stesso risultato, ma questo potrebbe non accadere se la tabella H non cambia

49) Data l'istanza di H(a,b,c,d,e) sotto a sinistra, la query alla sua destra:

a	b	c	d	e
a1	b4	c2	5	2021-01-20
a1	b2	c1	2	2020-12-02
a1	b3	c2	1	(NULL)
a1	b5	(NULL)	2	2020-01-03
a1	b1	c1	1	2020-03-01
a2	b3	c1	5	2021-05-28

```
SELECT a, LEAD(d,1) OVER(PARTITION BY a)  
FROM H;
```

- a) Non è errata sintatticamente ma usa la funzione LEAD in modo insensato
- b) Usa la funzione LEAD che si occupa di inserire nella current row il valore che il record successivo, avente lo stesso valore di 'd', assume su 'a'
- c) Nessuna alternativa è corretta
- d) Usa la funzione LEAD che si occupa di inserire nella current row il valore che il record precedente assume su 'd'
- e) È errata sintatticamente

# PRETEST

---

50) Si consideri la seguente base di dati e dire quale delle successive espressioni algebriche rappresenta l'interrogazione: "Trovare nome e cognome dei deputati della commissione di bilancio":

Deputato(codice, nome, cognome, commissione, provincia, collegio)  
Collegio(provincia, numero, nome)  
Provincia(sigla, nome, regione)  
Regione(codice, nome)  
Commissione(numero, nome, presidente)

a)

$\Pi_{\text{numero}}(\sigma_{\text{name}=\text{"Bilancio"}(\text{Commissione})}) \text{ join}_{\text{numero}=\text{commissione}} \Pi_{\text{name}, \text{cognome}, \text{commissione}} (\text{Deputato})$

b)

$\Pi_{\text{name}, \text{cognome}}(\Pi_{\text{numero}}(\sigma_{\text{name}=\text{"Bilancio"}(\text{Commissione})}) \text{ join}_{\text{numero}=\text{commissione}} \Pi_{\text{codice}, \text{name}, \text{cognome}, \text{commissione}} (\text{Deputato}))$

c) NESSUNA

d)

$\Pi_{\text{numero}}(\sigma_{\text{name}=\text{"Bilancio"}(\text{Commissione})}) \text{ join}_{\text{numero}=\text{commissione}} \Pi_{\text{codice}, \text{name}, \text{cognome}, \text{commissione}} (\text{Deputato})$

---

51) Si consideri la seguente base di dati e dire qual è il significato della successiva espressione algebrica:

Deputato(codice, nome, cognome, commissione, provincia, collegio)  
Collegio(provincia, numero, nome)  
Provincia(sigla, nome, regione)  
Regione(codice, nome)  
Commissione(numero, nome, presidente)

$\Pi_{\text{name}, \text{cognome}}(\Pi_{\text{presidente}}(\sigma_{\text{name}=\text{"Bilancio"}(\text{Commissione})}))$   
 $\bowtie_{\text{presidente}=\text{codice}}$   
 $\Pi_{\text{codice}, \text{name}, \text{cognome}} (\text{Deputato})$

- a) Trovare il nome e cognome del deputato con lo stesso nome del presidente della Commissione Bilancio
  - b) Trovare il nome e cognome dei deputati della Commissione Bilancio
  - c) Nessuna alternativa è corretta
  - d) Trovare il nome e cognome del presidente della Commissione Bilancio
-

# PRETEST

---

52) Si consideri la relazione  $R(A, \underline{B}, C, D)$ , quali delle seguenti proiezioni hanno certamente lo stesso numero di tuple di  $R$ . Scegli un'alternativa:

a) Nessuna alternativa è corretta

b)

$\pi_{AC}(R)$

c)

$\pi_{BD}(R)$

d)

$\pi_{CD}(R)$

---

53) Si consideri la seguente tabella, dire quale delle successive espressioni algebriche rappresenta le ricette che non contengono tabasco:

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

a)

$\Pi_{NomeRicetta}(Ricetta \triangleright \triangleleft Procedimento \triangleright \triangleleft \sigma_{Nome \neq Tabasco}(\text{Ingrediente}))$

b)

$\Pi_{NomeRicetta}(Ricetta \triangleright \triangleleft Procedimento \triangleright \triangleleft \sigma_{Nome=Tabasco}(\text{Ingrediente}))$

c)

$\Pi_{NomeRicetta}$   
-  $\Pi_{NomeRicetta}(Ricetta \triangleright \triangleleft Procedimento \triangleright \triangleleft \sigma_{Nome=Tabasco}(\text{Ingrediente}))$

# PRETEST

---

54) Si consideri la seguente tabella e la seguente espressione SQL. Dire quale delle successive espressioni algebriche calcola lo stesso risultato dell'espressione SQL:

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

```
SELECT NomeRicetta  
FROM Ricetta NATURAL JOIN Procedimento NATURAL JOIN Ingrediente  
WHERE CodIngr NOT IN ( SELECT CodIngr FROM Ingrediente WHERE Nome  
≠ 'zenzero')
```

a)

$\Pi_{NomeRicetta}(Ricetta \bowtie\lhd Procedimento \bowtie\lhd \sigma_{Nome='Zenzero'}(Ingrediente))$

b)

$\Pi_{NomeRicetta}(Ricetta \bowtie\lhd \sigma_{CodIngr='Zenzero'}(Procedimento))$

c)

$\Pi_{NomeRicetta}(Ricetta \bowtie\lhd Procedimento \bowtie\lhd \sigma_{Nome≠'Zenzero'}(Ingrediente))$

---

55) Si consideri la seguente tabella e la seguente espressione algebrica. Dire qual è il significato dell'espressione algebrica:

Deputato(codice, nome, cognome, commissione, provincia, collegio)  
Collegio(provincia, numero, nome)  
Provincia(sigla, nome, regione)  
Regione(codice, nome)  
Commissione(numero, nome, presidente)

$\pi_{nome, cognome}(\pi_{numero}(\sigma_{nome="Bilancio"}(Commissione)))$

$\bowtie_{commissione ≠ numero}$

$\pi_{commissione, nome, cognome}(Deputato))$

- a) Elencare i deputati membri di una commissione diversa dalla Commissione Bilancio
- b) Nessuna alternativa è corretta
- c) Elencare i membri della Commissione Bilancio
- d) Elencare i deputati non membri della Commissione Bilancio

# PRETEST

---

56) Si consideri la seguente tabella e la seguente espressione algebrica. Dire qual è il significato dell'espressione algebrica:

Deputato(codice, nome, cognome, commissione, provincia, collegio)  
Collegio(provincia, numero, nome)  
Provincia(sigla, nome, regione)  
Regione(codice, nome)  
Commissione(numero, nome, presidente)

$\Pi_{\text{nome}}(\Pi_{\text{codice}}(\sigma_{\text{nome}=\text{'Toscana'}}(\text{Regione}))$   
 $\bowtie_{\text{codice}=\text{regione}} \text{Provincia}$

- a) Nessuna alternativa è corretta
  - b) Tutti i nomi di province che hanno codice uguale a quello della Toscana
  - c) Tutti i nomi delle province
  - d) Tutti i nomi delle province della Toscana
- 

57) Si consideri la seguente tabella e la seguente espressione algebrica. Dire qual è il significato dell'espressione algebrica:

Deputato(codice, nome, cognome, commissione, provincia, collegio)  
Collegio(provincia, numero, nome)  
Provincia(sigla, nome, regione)  
Regione(codice, nome)  
Commissione(numero, nome, presidente)

$\Pi_{\text{nome}, \text{cognome}}(\sigma_{\text{commissione}=\text{null}}(\text{Deputato})) -$   
 $\Pi_{\text{nome}, \text{cognome}}($   
     $\Pi_{\text{numero}}(\sigma_{\text{nome}=\text{'esteri'}}(\text{Commissione}))$   
     $\bowtie_{\text{numero}=\text{commissione}}$   
     $\Pi_{\text{commissione}, \text{nome}, \text{cognome}}(\text{Deputato})$   
)

- a) Nessuna alternativa è corretta
  - b) Nome e cognome dei deputati non della commissione esteri
  - c) Nome e cognome dei deputati che stanno nella commissione esteri
  - d) Nome e cognome dei deputati che non sono in nessuna commissione
-

# PRETEST

---

58) Si consideri la seguente tabella e la seguente espressione nel calcolo relazionale dei domini. Dire qual è il significato dell'espressione nel calcolo relazionale dei domini:

Deputato(codice, nome, cognome, commissione, provincia, collegio)  
Collegio(provincia, numero, nome)  
Provincia(sigla, nome, regione)  
Regione(codice, nome)  
Commissione(numero, nome, presidente)

(nome:n, cognome:c | Commissione(numero: num, nome: nc, presidente:pr)  $\wedge$  Deputato(codice:pr, nome:n, cognome:c, commissione:num, provincia:pv, collegio:cg)  $\wedge$  nc='Bilancio')

- a) Trovare il nome e cognome del presidente della Commissione Bilancio
  - b) Elencare i nomi e cognomi dei deputati della Commissione Bilancio
  - c) Nessuna alternativa è corretta
  - d) Elencare i nomi e cognomi dei Deputati non nella Commissione Bilancio
- 

59) Data la seguente espressione del calcolo dei domini, dire qual è un'espressione algebrica corrispondente:

$R = \{ C:x, B:y \mid R(A:x) \wedge y=y \}$

a)

$P_{C \leftarrow A}(R \triangleright \neg B)$

b) Non ne esiste nessuna

c)

$P_{C \leftarrow A}(R)$

d)

$P_{C \leftarrow A}(R \sqcup B)$

# PRETEST

---

60) Si consideri la seguente tabella e la seguente espressione algebrica. Dire quale delle successive espressioni del calcolo dei domini rappresenta l'espressione algebrica:

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

$\Pi_{NomeRicetta} - \Pi_{NomeRicetta}(Ricetta \triangleright \triangleleft Procedimento \triangleright \triangleleft \sigma_{Nome=Tabasco}(Ingredient))$

a)

{NomeRicetta: nr | Ricetta(NomeRicetta: nr, ....)  $\wedge$   
Procedimento(Nome.ricetta:nr, CodIngr:ci)  $\wedge$  ci  $\neq$  "tabasco"}

b) Nessuna alternativa è corretta

c)

{NomeRicetta: nr | Ricetta(NomeRicetta: nr, ....)  $\wedge$   
Procedimento(NomeRicetta:nr, CodIngr: ci)  $\wedge$  Ingrediente(CodIngr:ci,  
Nome:n)  $\wedge$  Nome  $\neq$  "tabasco"}

---

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

1.  $\pi_{AB}(R) - \pi_{AB}(R \triangleright \triangleleft_{C=D} S)$   
2. {A:a, B:b | not exists e,c . R(A:a, B:b, C:c) and S(D:c, E:e) }

- a) Vero  
b) Falso
- 

62) Si consideri la seguente tabella e la seguente espressione nel calcolo relazionale dei domini. Dire qual è il significato dell'espressione nel calcolo relazionale dei domini:

Deputato(codice, nome, cognome, commissione, provincia, collegio)  
Collegio(provincia, numero, nome)  
Provincia(sigla, nome, regione)  
Regione(codice, nome)  
Commissione(numero, nome, presidente)

{nome:n, cognome:c | Commissione(numero: num, nome: nc, presidente:pr)  $\wedge$  Deputato(codice:cd, nome:n, cognome:c, commissione:num', provincia:pv, collegio:cg)  $\wedge$  nc='Esteri'  $\wedge$  num  $\neq$  num'}

- a) Elencare i deputati non presenti in commissione  
b) Elencare i deputati della Commissione Esteri  
c) Elencare i deputati presenti in commissioni diverse dalla Commissione Esteri
-

# PRETEST

---

63) Si consideri la seguente tabella e la seguente espressione nel calcolo relazionale dei domini. Dire qual è il significato dell'espressione nel calcolo relazionale dei domini:

Deputato(codice, nome, cognome, commissione, provincia, collegio)  
Collegio(provincia, numero, nome)  
Provincia(sigla, nome, regione)  
Regione(codice, nome)  
Commissione(numero, nome, presidente)

(nome:np | Provincia(sigla: sg, nome: np, regione: cr)  $\wedge$  Deputato(codice:cd, nome:n, cognome:c, commissione:num, provincia: sg, collegio: cg)  $\wedge$  Regione(codice: cr, nome:nr)  $\wedge$  (nr $\neq$ 'Sicilia'  $\wedge$  nr $\neq$ 'Toscana'))

- a) Elencare i nomi delle province non della Toscana né della Sicilia in cui c'è almeno un deputato eletto
  - b) Elencare i nomi delle province non della Toscana né della Sicilia senza deputati
  - c) Elencare i nomi delle province non della Toscana né della Sicilia
  - d) Nessuna alternativa è corretta
- 

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

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

- a) Vero
  - b) Falso
- 

65) Si considerino le relazioni **R1(A,B,C,D)**, **R2(B,D)**, **R3(A,C)**, l'espressione algebrica

$(R1 \bowtie R2) \cup (R1 \bowtie R3)$

è equivalente a quale delle seguenti espressioni nel calcolo delle tuple?:

- a) Non esiste un'espressione nel calcolo delle tuple corrispondente

b)

{g.(B,D) f.(A,C) | i(R1), g(R2), f(R3) | true}

- c) Nessuna alternativa è corretta

d)

{i.(A,B,C,D) | i(R1), g(R2), f(R3) | i.A=f.A and i.B=g.B and i.C=f.C and i.D=g.D}

# PRETEST

---

66) Si considerino le relazioni **R1(A,B,C,D)**, **R2(B,D)**, l'espressione algebrica

$$\rho_{E,F \leftarrow A,B}(\pi_{A,B,D}(R1 \bowtie R2))$$

è equivalente a quale delle seguenti espressioni nel calcolo delle tuple?:

a)

$$\{D:d, E:e, F:f \mid R1(A:e, B:f, C:c, D:d) \text{ and } R2(B:b, D:d) \}$$

b) Nessuna alternativa è corretta

c)

$$\{D:d, E:e, F:f \mid R1(E:e, F:f, C:c, D:d) \text{ and } R2(B:b, D:d) \}$$

d)

$$\{D:d, E:a, F:b \mid R1(A:a, B:b, C:c, D:d) \text{ and } R2(B:b, D:d) \}$$

---

67) Un DBMS relazionale:

- a) Può non essere transazionale
  - b) Non è transazionale
  - c) È sempre transazionale
- 

68) Le transazioni godono delle proprietà di:

- a) Nessuna alternativa è corretta
  - b) Atomicità, Consistenza, Isolamento, Persistenza
  - c) Atomicità, Persistenza, Durabilità, Conflict-serializzabilità
  - d) Consistenza, Isolamento, Persistenza, Recovery
  - e) Atomicità, Consistenza, Isolamento, Differibilità
- 

69) Adottando il protocollo **2PL stretto**:

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

# PRETEST

---

70) Una temporary table:

- a) Non può essere cancellata dall'utente, lo può fare solo il DBMS al termine della sessione
  - b) È temporanea, quindi viene distrutta dal DBMS al termine della query (o procedura) che la interroga
  - c) È cancellata dal DBMS al termine della sessione ed è subito ricreata e popolata all'inizio di una nuova sessione dello stesso utente
  - d) Nessuna alternativa è corretta
  - e) È cancellata dal DBMS dopo un tempo di inutilizzo che può essere impostato dall'utente
- 

71) In una materialized view **MV**, l'incremental refresh:

- a) Non è mai eseguito senza un'esplicita chiamata alla stored procedure
  - b) Copia il contenuto (tutto o parte) della log table in MV e cancella i record del log processati
  - c) Nessuna alternativa è corretta
  - d) Può essere di tipo immediate
- 

72) In una materialized view **MV**, l'incremental refresh:

- a) Può essere di tipo immediate, on demand, deferred
  - b) Copia tutto il contenuto della log table in MV, e poi cancella il contenuto della log table
  - c) È sempre seguito tramite una chiamata a una stored procedure
  - d) Copia tutto il contenuto (o parte del contenuto) della log table in MV, e poi cancella dalla log table i record copiati
  - e) Nessuna alternativa è corretta
- 

73) Quasi tutte le implementazioni di DMBS usano architetture distribuite. Nell'architettura client-server, il server:

- a) Non necessita di un'interfaccia dei servizi messi a disposizione, nell'interazione col client
  - b) Ha un ruolo attivo perché genera richieste
  - c) Nessuna alternativa è corretta
  - d) Può non essere unico
- 

74) I database relazionali rispetto a quelli NoSQL sono:

- a) Meno affidabili e meno scalabili
  - b) Meno affidabili e più scalabili
  - c) Più affidabili e meno scalabili
  - d) Più affidabili e più scalabili
-

# PRETEST

---

75) La ripresa a freddo viene applicata quando c'è un guasto ad un dispositivo:

- a) Se tutte le transazioni cominciate prima dell'ultimo checkpoint sono terminate con successo prima del guasto
  - b) Sempre
  - c) Se nessuna transazione è abortita tra l'ultimo checkpoint e il guasto
  - d) Se nessuna transazione è cominciata dopo l'ultimo checkpoint
- 

76) Prima di inserire un record di checkpoint nel log, il gestore dell'affidabilità:

- a) Nessuna alternativa è corretta
  - b) Rifiuta nuovi begin-transaction e nuovi commit/abort
  - c) Non può rifiutare nuovi begin-transaction né aspettare che tutte le transazioni iniziate eseguano commit o abort
  - d) Può rifiutare nuovi begin-transaction e aspettare che tutte le transazioni iniziate eseguano commit o abort
- 

77) Una variabile user-defined usata all'interno di una stored procedure:

- a) È visibile solo all'interno della stored procedure
  - b) Non ha bisogno di essere dichiarata
  - c) Nessuna alternativa è corretta
  - d) Può essere letta e scritta ovunque, ma gli effetti di tali operazioni sono cancellati al termine della stored procedure
- 

78) Una query con raggruppamento:

- a) Può contenere nel group by anche attributi che non sono presenti nella select
  - b) Restituisce sempre un numero di record minore della tabella su cui si esegue perché il group by prevede sempre l'uso di operatori di aggregazione
  - c) Nessuna alternativa è corretta
  - d) Non può mai contenere nel group by attributi che non sono presenti anche nel select
- 

79) Esistendo un gestore della concorrenza dei DBMS relazionali, una transazione non risente degli effetti di altre transazioni concorrenti, quindi la transazione:

- a) Nessuna alternativa è corretta
  - b) Non espone i suoi stati intermedi solo quando il controllore dell'affidabilità recupera uno stato consistente della base di dati dopo un guasto
  - c) Non espone mai i suoi stati intermedi
  - d) Può esporre i suoi stati intermedi perché vengono garantiti essere corretti
-

# PRETEST

---

80) I database NoSQL permettono di realizzare tabelle:

- a) Solo con tuple di dimensione fissa
  - b) Con tuple di dimensione qualsiasi
  - c) Con tuple con elementi tutti dello stesso tipo
  - d) Nessuna alternativa è corretta
- 

81) Il gestore della concorrenza dei DBMS:

- a) Accetta le richieste di operazioni di lettura/scrittura provenienti dalle transazioni, poi prima di farle terminare, verifica se lo schedule ottenuto è serializzabile
  - b) Accetta le richieste di operazioni di lettura/scrittura provenienti dalle transazioni solo se le transazioni lavorano su oggetti distinti, altrimenti le fa abortire
  - c) Non segue nessuno dei comportamenti descritti nelle altre alternative
  - d) Accetta le richieste di operazioni di lettura/scrittura provenienti dalle transazioni, poi verifica se lo schedule ottenuto è serializzabile solo se tutte le transazioni sono pronte a terminare con successo
- 

82) Se due schedule sono tra loro view-equivalenti:

- a) Non possono mai essere eseguiti usando il protocollo time-stamp
  - b) Nessuna alternativa è corretta
  - c) Possono essere view-serializzabili
  - d) Non possono essere conflict-serializzabili
- 

83) Due espressioni algebriche:

- a) Sono equivalenti se producono un risultato con lo stesso schema
  - b) Sono equivalenti se producono un risultato con un numero di tuple uguale
  - c) Sono equivalenti se producono un risultato con lo stesso schema e le stesse tuple
- 

84) Il gestore dell'affidabilità dei DBMS relazionali ha il compito di:

- a) Garantire che gli effetti di una transazione andata in commit non vadano perduti
  - b) Nessuna alternativa è corretta
  - c) Garantire che l'effetto di una interrogazione sia corretto
  - d) Permettere di recuperare i dati modificati dalle transazioni interrotte a seguito di un guasto
-

# PRETEST

---

85) Dato un insieme **F** di dipendenze funzionali, le regole di Armstrong servono per:

- a) Eliminare le dipendenze banali da **F**
  - b) Verificare se le dipendenze di **F** sono o no banali
  - c) Derivare tutte le dipendenze implicate da **F**
  - d) Nessuna alternativa è corretta
- 

86) Indicare la complessità dell'operazione di verificare se uno schema **R** soddisfa la **3FN**:

- a) Costante
  - b) Polinomiale
  - c) Esponenziale
- 

87) Una transazione comprende una o più operazioni di accesso alla base di dati, e queste operazioni sono:

- a) Inserzioni, cancellazioni, modifiche e interrogazioni
  - b) Non interrogazioni
  - c) Solo cancellazioni
  - d) Solo inserzioni
- 

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

- a) È un prodotto cartesiano
  - b) Estende, con valori NULL, le ennuple che verrebbero escluse e le mantiene
  - c) È un join incompleto
  - d) Nessuna alternativa è corretta
  - e) È un join vuoto
- 

89) Considerate le analytic function **rank** e **dense\_rank**:

- a) Data una tabella con N record, la **dense\_rank** numera i record usando sempre tutti i valori in (1,...,N), mentre la **rank** numera i record usando un sottoinsieme di (1,...,N) perché non produce duplicati
  - b) Nessuna alternativa è corretta
  - c) Prendono come argomento un attributo della tabella (a valori numerici) in base al quale stilano una classifica dei record
  - d) Se usate l'una al posto dell'altra, senza modificare il resto del codice della query, non restituiscono mai lo stesso risultato
  - e) Si possono usare solo in query con raggruppamento
-

# PRETEST

---

90) L'indice secondario di un file è un file denso:

- a) Nessuna alternativa è corretta
  - b) Sempre
  - c) Mai
- 

91) Se uno schedule è eseguibile col protocollo **2PL**, allora tale schedule:

- a) Non è mai view-serializzabile
  - b) Non è mai conflict-serializzabile
  - c) Non è mai eseguibile con il time stamp
  - d) Nessuna alternativa è corretta
- 

92) Due transazioni possono essere eseguite concorrentemente solo se:

- a) Hanno time stamp uguali
  - b) Nessuna delle condizioni espresse nelle altre alternative è una condizione necessaria
  - c) Non contengono operazioni di scrittura
  - d) Lavorano su oggetti differenti
- 

93) La semantica di SQL è data mediante:

- a) Il calcolo dei domini
  - b) Il calcolo delle tuple
  - c) L'algebra relazionale
  - d) Nessuna alternativa è corretta
- 

94) L'effetto delle singole operazioni di lettura e scrittura interne a una transazione deve essere visibile all'esterno:

- a) Solo se la transazione termina con successo
  - b) In nessuno dei due casi descritti
  - c) Solo quando la transazione fallisce, per individuare il motivo del fallimento
- 

95) Il calcolo relazionale ammette espressioni dipendenti dal dominio degli attributi, mentre l'algebra relazionale non le ammette mai:

- a) Vero
  - b) Falso
-

# PRETEST

---

96) Essere ‘prima o poi consistente’ è una caratteristica dei database:

- a) NoSQL
  - b) Di nessun tipo
  - c) Relazionali
- 

97) I database relazionali offrono supporto al concetto di transazione:

- a) Mai
  - b) Qualche volta
  - c) Dipende dal tipo di applicazione che devono gestire
  - d) Sempre
- 

98) Se uno schedule è **CSR** allora:

- a) È eseguibile solo usando il protocollo 2PL stretto
  - b) È anche eseguibile secondo il protocollo 2PL
  - c) Non è detto che sia eseguibile né col protocollo 2PL né con il 2PL stretto
- 

99) I vincoli di integrità referenziale:

- a) Nessuna alternativa è corretta
  - b) Permettono di collegare i valori di un attributo di una tabella con quelli di un attributo di una tabella necessariamente diversa
  - c) Permettono di collegare i valori di un attributo di una tabella con quelli di un attributo di un’altra tabella, o della stessa tabella
  - d) Si possono definire solo in presenza di chiavi esterne
- 

100) La primitiva **force** viene usata dal gestore del buffer per:

- a) Nessuna alternativa è corretta
  - b) Trasferire dati dalla memoria secondaria al buffer in assenza di posizioni vuote nel buffer stesso
  - c) Portare immediatamente i risultati di una transazione che ha effettuato commit, dal buffer alla memoria secondaria
  - d) Trasferire dati dalla memoria secondaria a seguito di un guasto software
  - e) Trasferire dati dalla memoria secondaria a seguito di un guasto hardware
-

# PRETEST

---

101) 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:

- a) Falso
  - b) Vero
- 

102) Dato uno schedule eseguito con il protocollo **S2PL**:

- a) Può esistere un solo schedule conflict-equivalente ad esso
  - b) Non esiste alcuno schedule seriale view-equivalente ad esso
  - c) Esiste almeno uno schedule view-serializzabile equivalente ad esso
  - d) Nessuna alternativa è corretta
- 

103) Una temporary table:

- a) Deve essere sempre cancellata dall'utente prima del termine della sessione tramite una DROP
  - b) È temporanea, cioè viene eliminata dal DMBS al termine della query (o stored program) che la interroga
  - c) Il suo contenuto non può essere aggiornato direttamente come le table classiche, ma occorre sempre fare una DROP e poi una nuova CREATE
  - d) Può essere cancellata dall'utente, oppure dal DMBS
  - e) Nessuna alternativa è corretta
- 

104) Una volta creata, una stored function:

- a) Nessuna alternativa è corretta
  - b) Può essere sempre chiamata nel codice dichiarativo, eccetto che nel FROM e nel GROUP BY
  - c) Può essere sempre chiamata tramite una CALL, in qualsiasi parte del codice, sia dichiarativo che procedurale
  - d) Può essere richiamata anche all'interno di codice procedurale, ma nel caso restituisca più record, il suo risultato deve essere assegnato a un cursore
- 

105) I record di log sono scritti dopo che i corrispondenti record sono stati scritti nella base di dati:

- a) Falso
  - b) Vero
-

# PRETEST

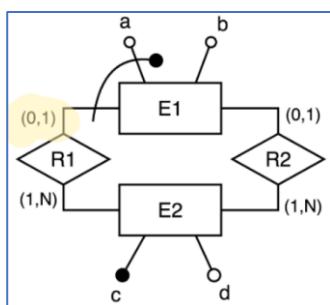
---

106) Considerare lo schema  $R(X)$  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, la seguente uguaglianza: (scegli un'alternativa)

$$|\pi_Y(r)| = |r|$$

- a) È vera
  - b) Non si può dire a priori se è vera o falsa
  - c) È falsa
- 

107) 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
- 

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

- a) Vero
  - b) Falso
- 

109) Una base di dati ottenuta da un processo di progettazione:

- 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
- 

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

- a) Falso
  - b) Vero
-

# PRETEST

---

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

- a) Vero
  - b) Falso
- 

112) Il modello relazionale:

- 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
- 

113) Il **protocollo 2PL stretto**:

- a) Aumenta la concorrenza tra le transazioni perché permette di acquisire lock solo quando servono
  - b) Nessuna alternativa è corretta
  - c) Aumenta la concorrenza tra le transazioni perché riduce il tempo in cui viene posto un lock a una risorsa
  - d) Elimina il fenomeno delle letture sporche
- 

114) In una materialized view **MV** gestita tramite incremental refresh in modalità deferred:

- a) Non fa uso di log table, ma usa un event che aggiorna MV
  - b) Il refresh non può essere complete perché il livello di aggiornamento ottenibile dipende dall'istante di esecuzione dell'event
  - c) La log table parzialmente svuotata anche se si esegue il refresh in modalità complete, perché l'istante di esecuzione dell'event determina i record da cancellare dalla log table
  - d) Deve essere fissato un numero massimo di row della log table, dopodichè si deve procedere al refresh
  - e) Nessuna alternativa è corretta
- 

115) Se uno schedule è eseguibile con il protocollo **time stamp**:

- a) Può esistere più di uno schedule seriale equivalente
  - b) Nessuna alternativa è corretta
  - c) Esiste un unico schedule seriale corrispondente
  - d) Può non corrispondere ad alcuno schedule seriale equivalente
-

# PRETEST

---

116) Il calcolo delle tuple e il calcolo dei domini possono essere usati l'uno al posto dell'altro per specificare una qualsiasi interrogazione:

- a) Solo per domini numerici
  - b) Solo quando non è necessario l'uso dell'operatore di unione
  - c) In tutti i casi
- 

117) Due espressioni algebriche equivalenti hanno lo stesso costo in esecuzione:

- a) Nessuna alternativa è corretta
  - b) Solo se non contengono selezioni
  - c) Mai
  - d) Sempre
- 

118) In una stored procedure:

- a) Per uscire da un ciclo deve essere sempre dichiarato e usato un handler di tipo continue che gestisca la condizione NOT FOUND
  - b) Nessuna alternativa è corretta
  - c) Per uscire da un ciclo può essere necessario un handler
  - d) L'unico modo per uscire da un ciclo è usare una SIGNAL
- 

119) L'architettura distribuita a 3 livelli per il DBMS separa le applicazioni:

- a) Dal client
  - b) Sia dal client che dal server
  - c) Dal database server
- 

120) Una relazione uno a uno fra due entità **E1** ed **E2**:

- a) Ammette più di un tipo di traduzione nel modello logico relazionale
  - b) È utile solo se E1 ed E2 coincidono
  - c) Se E1 o E2 partecipano opzionalmente, tradurre la relazione con una tabella può introdurre valori NULL nella tabella
  - d) Se E1 o E2 partecipano opzionalmente, non si può tradurre con una tabella
  - e) Serve per esprimere il concetto che, in un certo istante, un'occorrenza di E1 è in relazione con un'occorrenza di E2, tuttavia, nel tempo, l'occorrenza di E1 può essere legata a più occorrenze di E2 contemporaneamente
-

# PRETEST

---

121) Data una tabella **T(a,b,c,d)** e considerata una query con raggruppamento che contiene una condizione sui gruppi del tipo “**HAVING aggr(d)=....**” dove **aggr(d)** è una funzione di aggregazione, l’uso di *distinct* nel select:

- a) Può essere utile per eliminare i duplicati solo se, fra gli attributi di raggruppamento, non è presente la chiave di T, cioè la coppia di attributi ‘a’ e ‘b’
  - b) Nessuna alternativa è corretta
  - c) Non ha mai senso usare distinct nel SELECT
  - d) Può essere utile per eliminare i duplicati se, raggruppando su ‘a’ e ‘b’, si proietta anche aggr(d)
  - e) È inutile se, fra gli attributi di raggruppamento, è presente la chiave di T, cioè la coppia di attributi ‘a’ e ‘b’, indipendentemente da cosa si proietta
- 

122) I database NoSQL scalano nello stesso modo dei database relazionali:

- a) Falso
  - b) Vero
- 

123) Le letture sporche non possono verificarsi tra due transazioni concorrenti a meno che una delle due abortisca:

- a) Falso
  - b) Vero
- 

124) Quando il gestore dell'affidabilità decide di inserire un record di checkpoint nel log:

- a) Blocca completamente l'esecuzione delle transazioni
  - b) Può continuare ad accettare tutte le operazioni richieste dalle transazioni compresi i commit ma non gli abort
  - c) Nessuna delle alternative precedenti
  - d) Non accetta più commit o abort dalle transazioni
- 

125) Al momento di inserire un checkpoint nel log il DBMS può rifiutare nuovi commit o aspettare i commit di tutte le transazioni iniziate: al momento del recovery in entrambi i casi:

- a) Esiste sia una lista di transazioni UNDO che REDO
  - b) Si producono situazioni differenti che richiedono comportamenti differenti
  - c) Esiste solo la lista REDO
- 

126) L'indice primario di un file può essere:

- a) Solo sparso
  - b) Solo denso
  - c) Sia denso che sparso
-

# PRETEST

---

127) La scrittura nel database del risultato di un'operazione può essere fatta:

- a) Necessariamente prima del commit della transazione
  - b) In qualunque momento
  - c) Necessariamente dopo il commit della transazione
- 

128) Date due relazioni  $R(A,B)$  e  $S(C,B,A)$ , la seguente espressione:

$$R \bowtie R.B = S.B \quad S$$

produce lo stesso risultato dell'espressione:

$$R \bowtie S$$

- a) Dipende
  - b) Sempre
  - c) Mai
- 

129) Una relazione non in **3NF** può essere portata in **3NF** senza perdite sul join della memoria secondaria:

- a) Falso
  - b) Vero
- 

130) Una stored procedure viene cancellata:

- a) Mai
  - b) Al termine della sessione e in una nuova sessione va eseguito nuovamente il CREATE PROCEDURE
  - c) Dopo che ha restituito il risultato
  - d) Su richiesta dell'utente
  - e) Nessuna delle risposte è corretta
- 

131) Un trigger **BEFORE INSERT** sulla tabella **T**:

- a) Non è detto che sia eseguito a seguito dell'esecuzione di ogni statement di tipo INSERT su T
  - b) Non è eseguito se la tabella è vuota
  - c) Va in esecuzione prima che l'utente esegua uno statement di tipo INSERT sulla tabella T
  - d) Nessuna alternativa è corretta
-

# PRETEST

---

132) Nella ripresa a caldo, dipendentemente dallo stato del log, tutte le operazioni tra il dump e l'ultimo checkpoint:

- a) Non possono essere né rifatte né disfatte
  - b) Possono essere rifatte ma non disfatte
  - c) Possono essere solo disfatte
  - d) Possono essere disfatte o rifatte
- 

133) Il concetto di transazione è:

- a) Specifico dei database relazionali
  - b) Nessuna alternativa è corretta
  - c) Specifico dei database NoSQL
  - d) Comunque sia ai database relazionali che NoSQL
- 

134) Il fenomeno delle letture sporche:

- a) Può essere gestito al momento della gestione di un guasto
  - b) Non può essere gestito in nessun modo
  - c) Può essere gestito con un aborto in cascata tra le transazioni
  - d) Nessuna alternativa è corretta
- 

135) I record di log sono scritti:

- a) Dopo che i corrispondenti record sono stati scritti nella base di dati
  - b) Nessuna alternativa è corretta
  - c) Solo dopo che la transazione ha effettuato il commit
  - d) Prima che i corrispondenti record sono stati scritti nella base di dati
- 

136) Una stored function deterministic:

- a) Date due esecuzioni con due insiemi di valori dei parametri d'ingresso, tali esecuzioni producono sempre due uscite diverse
  - b) Al variare dei valori dei parametri d'ingresso restituisce sempre lo stesso risultato
  - c) Nessuna alternativa è corretta
  - d) Non restituisce lo stesso risultato se, in esecuzioni diverse, i valori dei parametri d'ingresso non cambiano
  - e) Può restituire risultati diversi se, in esecuzioni diverse, i valori dei parametri d'ingresso cambiano
-

# PRETEST

---

137) I database NoSQL

- a) Sono basati sulle proprietà ACID
  - b) Distribuiscono i dati su più server, alcuni dei quali non contengono dati sempre consistenti con quelli degli altri server
  - c) Nessuna alternativa è corretta
  - d) Sono ‘eventual consistent’, cioè esiste l’eventualità che siano consistenti, ma non esiste una garanzia che i dati siano o saranno consistenti
- 

138) I vincoli di integrità di una base di dati vengono valutati dal DBMS:

- a) Solamente prima di ogni tentativo di modifica di dati contenuti in tabelle interessate dai vincoli
  - b) Prima di ogni tentativo di inserzione di un nuovo elemento in una tabella interessata dai vincoli
  - c) Prima di ogni tentativo di modifica di dati contenuti in tabelle interessate dai vincoli o di inserzione di nuovi elementi in tali tabelle
  - d) Ad ogni tentativo di lettura dei dati in tabelle interessate dai vincoli
- 

139) Cos’è il fattore di blocco

- a) La dimensione di un blocco nella memoria secondaria
  - b) La parte intera del rapporto tra la dimensione del blocco e quella del record di una relazione
  - c) Il rapporto tra la dimensione del blocco e quella del record di una relazione
- 

140) Si consideri la relazione  $R(A,B,C)$ . Sapendo che il numero  $N(R)$  di tuple di  $R$  è 50000, il numero di valori diversi dell’attributo  $C$  è 100, il numero di valori diversi dell’attributo  $A$  è 50, e supponendo equiprobabile la distribuzione dei valori  $C$  ed  $A$ , qual è il costo in termini della dimensione dei risultati intermedi delle seguenti relazioni:

$$R1 = \sigma_{C=1}^{100} ( \sigma_{A=1}^5 (R))$$

$$R2 = \sigma_{A=1}^5 ( \sigma_{C=1}^{100} (R))$$

- a) Nessuna alternativa è corretta
  - b) R1 richiede una tabella di 500 elementi, R2 di 1000 elementi
  - c) R1 richiede una tabella di 1000 elementi, R2 di 500 elementi
  - d) R1 e R2 richiedono tabelle della stessa dimensione
-

# PRETEST

---

141) Si consideri la relazione  $R(A,B,C)$ : il numero  $N(R)$  di tuple di  $R$  è 50000. Il numero di valori diversi degli attributi  $A$ ,  $B$ ,  $C$  è 100. Supponendo equiprobabile la distribuzione dei valori  $C$ , qual è la dimensione del risultato della seguente espressione:

$\pi_{AB}(\sigma_{C='n1'}(R))$

- a) 500
  - b) 10000
  - c) 5000
- 

142) Si consideri la relazione  $R(A,B,C)$ : il numero  $N(R)$  di tuple di  $R$  è 50000. Il numero di valori diversi degli attributi  $A$ ,  $B$  è 50 e di  $C$  è 10000. Supponendo equiprobabile la distribuzione dei valori  $C$ , qual è la dimensione del risultato della seguente espressione:

$\pi_{AB}(\sigma_{C='n1'}(R))$

- a) 2500
  - b) 10000
  - c) 5
- 

143) Si consideri la relazione  $R(A,B,C)$ : il numero  $N(R)$  di tuple di  $R$  è 50000. Il numero di valori diversi degli attributi  $A$ ,  $B$  è 50 e di  $C$  è 10000. Supponendo equiprobabile la distribuzione dei valori  $C$ , qual è la dimensione del risultato della seguente espressione:

$\pi_C(\sigma_{A='n1'}(R))$

- a) 1000
  - b) 10000
  - c) Nessuna alternativa è corretta
  - d) 50
- 

144) Si consideri la relazione  $R(A,B,C)$ : il numero  $N(R)$  di tuple di  $R$  è 50000. Il numero di valori diversi degli attributi  $A$ ,  $B$ ,  $C$  è 100. Supponendo equiprobabile la distribuzione dei valori  $C$ , qual è la dimensione del risultato della seguente espressione:

$(\sigma_{A='n1'}(\pi_{AB}(R)))$

- a) Nessuna alternativa è corretta
  - b) 500
  - c) 100
  - d) L'espressione è sbagliata sintatticamente
-

# PRETEST

145) I database transazionali:

- a) Garantiscono che prima o poi la consistenza venga ottenuta, sebbene il database possa essere in uno stato inconsistente durante il suo utilizzo
- b) Garantiscono le cosiddette proprietà ACID
- c) Garantiscono che la persistenza venga mantenuta a spese della consistenza
- d) Nessuna alternativa è corretta

146) Si consideri la seguente base di dati, qual è il significato della successiva espressione nel calcolo relazionale dei domini? :

Deputato(codice, nome, cognome, commissione, provincia, collegio)  
Collegio(provincia, numero, nome)  
Provincia(sigla, nome, regione)  
Regione(codice, nome)  
Commissione(numero, nome, presidente)

{nome:n, cognome:c | Deputato(codice:cd, nome:n, cognome:c, commissione:num, provincia: sg, collegio: cg)  $\wedge$  not exists nm, nc .  
Commissione(numero:nm, nome:nc, presidente:cd)}

- a) Elenca i nomi e congnomi dei deputati
- b) Elenca i nomi e congnomi dei deputati che non sono presidenti di commissione
- c) Elenca i nomi e congnomi dei deputati che sono presidenti di commissione
- d) Nessuna alternativa è corretta

147) Una query con raggruppamento:

- a) Raggruppa sempre su un sottoinsieme degli attributi presenti nel select
- b) Restituisce sempre almeno un record perché ogni gruppo esiste sempre
- c) Nessuna alternativa è corretta
- d) Restituisce tanti record quanti sono i gruppi non eliminati dalla clausola where

148) Data l'istanza T(a,b,c,d,e) sotto a sinistra, la query alla sua destra:

T				
a	b	c	d	e
a1	b1	4	2021-01-07	3
a1	b2	3	2022-05-20	3
a2	b3	4	2022-03-30	4
a3	b1	1	2021-05-20	(NULL)
a3	b2	2	2022-02-22	2
a4	b3	1	2022-06-01	1
a5	b3	2	2021-05-18	3

```
SELECT T.a, T.b,  
       RANK() OVER(PARTITION BY T.e  
                   ORDER BY T.c) as r  
FROM T  
WHERE T.e > 1;
```

- a) Restituisce una classifica con pari merito
- b) Restituisce una classifica che non contiene pari merito, anche se cambia l'istanza di T
- c) Nessuna alternativa è corretta
- d) Restituisce una classifica senza pari merito

# PRETEST

---

149) Un evento di tipo *recurring*:

- a) Nessuna alternativa è corretta
  - b) Può non contenere le clausole *STARTS* e *ENDS*
  - c) Deve contenere le clausole *STARTS* e *ENDS*
  - d) Deve obbligatoriamente contenere la clausola *STARTS*, altrimenti non andrà mai in esecuzione
- 

150) Uno schedule **view serializzabile** può essere equivalente ad uno risultante dall'esecuzione delle richieste delle transazioni sulla base del protocollo **time-stamp**:

- a) Sempre
  - b) In nessun caso
  - c) Solo se non è view equivalente a uno schedule ottenibile usando il protocollo 2PL
  - d) Nessuna alternativa è corretta
- 

151) Si consideri il seguente schema di base di dati, specificare qual è il significato della successiva espressione algebrica:

Prenotazioni (CodFiscaleCliente, DataChiamata, OraChiamata, IDPrenotazione, NumOspitiPrenotazione, NumStanzePrenotate, DataInizioPrenotazione, DataFinePrenotazione)  
Cliente(NomeCliente, CognomeCliente, CodFiscaleCliente, RecapitoCliente)  
Stanza (NumStanza, CategoriaStanza, NumPianoStanza, NumLettiMaxStanza)  
Servizi (NumStanza, IDTariffa, ServiziStanza, CostoStanza)  
Tariffa (IDTariffa, DataInizioTariffa, DataFineTariffa)  
PrenotazioneStanza(IDPrenotazione, NumStanza)

```
πNumStanza, IDTariffa(  
    ( πDataInizioPrenotazione, DataFinePrenotazione, IDPrenotazione(σCodFiscale='XXX' (Prenotazioni))  
        ⋙ πNumStanza(PrenotazioneStanza)  
    ) ⋙ DataInizioPrenotazione >= DataInizioTariffa and DataFinePrenotazione <= DataFineTariffa  
    Tariffa  
)
```

- a) Produce l'elenco delle stanze associate con l'identificatore della tariffa di tutte le prenotazioni effettuate da 'XXX'
  - b) Nessuna alternativa è corretta
  - c) Produce l'elenco delle stanze associate con l'identificatore della tariffa delle prenotazioni effettuate da 'XXX', per periodi interamente contenuti nel periodo di durata di una tariffa
  - d) Produce l'elenco delle stanze libere per ogni tariffa
-

# PRETEST

---

152) Sia data l'istanza di **T(a,b,c,d,e)** sotto a sinistra. Supponendo sia stata creata (vuota) una materialized view **M(a,tot)**, dove tot è pari alla somma dei valori di c per ogni valore di a in **T**, il codice alla sua destra:

T				
a	b	c	d	e
a1	b1	4	2021-01-07	3
a1	b2	3	2022-05-20	3
a2	b3	4	2022-03-30	4
a3	b1	1	2021-05-20	(NULL)
a3	b2	2	2022-02-22	2
a4	b3	1	2022-06-01	1
a5	b3	2	2021-05-18	3

```

INSERT INTO M
SELECT T.a, SUM(T.c)
FROM T
GROUP BY T.a;
CREATE TRIGGER tt
AFTER INSERT ON T FOR EACH ROW
    UPDATE M
        SET M.tot = M.tot + NEW.c
        WHERE M.a = NEW.a;

```

- a) Nessuna alternativa è corretta
  - b) Effettua il build di M, dopodichè la mantiene in sync con i raw data
  - c) Effettua il build di M, e implementa l'immediate refresh di M
  - d) È errato sintatticamente (non compila)
- 

153) Si consideri la seguente base di dati, dire qual è il significato della successiva espressione nel calcolo relazionale dei domini?:

```

Deputato(codice, nome, cognome, commissione, provincia, collegio)
Collegio(provincia, numero, nome)
Provincia(sigla, nome, regione)
Regione(codice, nome)
Commissione(numero, nome, presidente)

```

```

{nome:n, cognome:c | Deputato(codice:cd, nome:n, cognome:c, commissione:num, provincia: sg, collegio: cg) ∧
not exists num, sg, cg . Deputato(codice:cd', nome:n, cognome:c, commissione:num, provincia: sg, collegio: cg)
∧ cd ≠ cd'}

```

- a) Elenca i nomi e i cognomi di tutti i deputati
- b) Elenca i nomi e i cognomi dei deputati che hanno omonimi
- c) Elenca i nomi e i cognomi dei deputati che non hanno omonimi
- d) Nessuna alternativa è corretta

# PRETEST

---

154) Considerare il diagramma E-R e la tabella dei volumi sottostanti. Supponendo che i valori di **a1** siano **5** ed equiprobabili, indicare il numero di accessi necessari, nel caso peggiore, per restituire, per ciascuna occorrenza di **A** tale che **a1='xxx'** o **a1='yyy'**, il codice **ka** dell'occorrenza, e il numero di valori diversi di **c** a cui essa è associata:

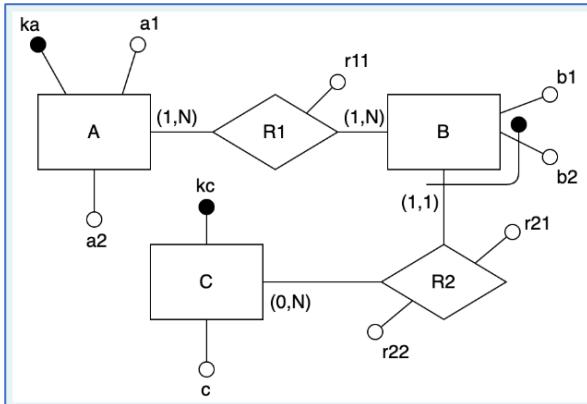


Tavola dei volumi	
A	1000
B	500
C	50
R1	4000

- a) 4750
  - b) Nessuna alternativa è corretta
  - c) 4050
  - d) 5550
- 

155) Si consideri la relazione **R(A,B,C)**; il numero **N(R)** di tuple di **R** è **50000**, il numero di valori diversi degli attributi **A** e **B** è di **500**, e di **C** è **1000**; ciascun elemento di **A**, **B** e **C** occupa una parola di memoria. Supponendo equiprobabili le distribuzioni dei valori degli attributi, qual è la dimensione in termini di parole occupate dal risultato della seguente espressione nel caso peggiore?:

$\Pi_{C,A} (\sigma_{A='n1' \text{ and } B='n2'}(R))$

- a) 200
  - b) 2000
  - c) 500
  - d) Nessuna alternativa è corretta
- 

156) Ogni insieme minimale di dipendenze:

- a) Nessuna alternativa è corretta
  - b) Non è mai in forma normale
  - c) Può essere in forma normale
  - d) Se non è in forma normale, non può essere normalizzato
-

# PRETEST

---

157) La query sotto a destra eseguita sull'istanza della tabella **T** sotto a sinistra:

T				
a	b	c	d	e
a1	b1	4	2021-01-07	3
a1	b2	3	2022-05-20	3
a2	b3	4	2022-03-30	4
a3	b1	1	2021-05-20	(NULL)
a3	b2	2	2022-02-22	2
a4	b3	1	2022-06-01	1
a5	b3	2	2021-05-18	3

```
SELECT T.*  
FROM T  
WHERE T.a <> 'a3' OR T.b <> 'b1';
```

- a) Elimina due record
  - b) Non elimina alcun record
  - c) Elimina un record
  - d) Nessuna alternativa è corretta
- 

158) Si consideri il seguente insieme di dipendenze **AB→C, B→A, C→E, AB→E** per la relazione **R**, qual è un insieme minimale equivalente?:

- a)  $AB \rightarrow C, C \rightarrow E, AB \rightarrow E$
  - b)  $B \rightarrow C, B \rightarrow A, C \rightarrow E$
  - c) L'insieme è già minimale
  - d)  $AB \rightarrow C, B \rightarrow A, C \rightarrow E$
- 

159) Uno schedule **view serializzabile** può essere view equivalente ad uno risultante dall'esecuzione delle transazioni sulla base del protocollo **2PL stretto**:

- a) Solo se è view equivalente a uno schedule ottenibile usando il protocollo time-stamp
  - b) Ma deve essere almeno anche conflict serializzabile
  - c) Nessuna alternativa è corretta
  - d) Solo
  - e) Non è view equivalente allo schedule che riproduce l'ordine delle richieste delle transazioni
- 

160) Si consideri la relazione **R(A,B,C)**: il numero **N(R)** di tuple di **R** è **50000**, il numero di valori diversi degli attributi **A** e **B** è **100**, e di **C** è **1000**. Supponendo equiprobabile la distribuzione dei valori di **C**, qual è la dimensione del risultato, in termini di numero di tuple, della seguente espressione nel caso peggiore?:

$$\pi_{A,B}(\sigma_{C='n1' \text{ or } C='n2'}(R))$$

- a) 100
  - b) 1000
  - c) Nessuna alternativa è corretta
  - d) 50
-

# PRETEST

161) Data una stored procedure, è possibile convertirla in una stored function che ottiene lo stesso risultato?:

- a) Sì, sempre
- b) No, non è mai possibile
- c) Nessuna alternativa è corretta
- d) Sì, ma non sempre

162) Si consideri il seguente schema di base di dati e dire qual è il significato della successiva espressione algebrica:

```
Prenotazioni (CodFiscaleCliente, DataChiamata, OraChiamata, IDPrenotazione,  
    NumOspitiPrenotazione, NumStanzePrenotate, DataInizioPrenotazione, DataFinePrenotazione)  
Cliente (NomeCliente, CognomeCliente, CodFiscaleCliente, RecapitoCliente)  
Stanza (NumStanza, CategoriaStanza, NumPianoStanza, NumLettiMaxStanza)  
Servizi (NumStanza, IDTariffa, ServiziStanza, CostoStanza)  
Tariffa (IDTariffa, DataInizioTariffa, DataFineTariffa)  
PrenotazioneStanza(IDPrenotazione, NumStanza)
```

```

$$\pi_{\text{NumStanza}, \text{IDTariffa}} ( \pi_{\text{DataInizioPrenotazione}, \text{IdPrenotazione}} (\sigma_{\text{CodFiscale}='XXX'} (\text{Prenotazioni})) \bowtie \text{PrenotazioneStanza} ) \bowtie \text{Tariffa} \text{ such that } \text{DataInizioPrenotazione} \geq \text{DataInizioTariffa} \text{ and } \text{DataInizioPrenotazione} \leq \text{DataFineTariffa}$$

```

- a) Produce l'elenco delle stanze associate con l'identificatore della tariffa di tutte le prestazioni effettuate da 'XXX', se la prenotazione rientra completamente in una sola tariffa
- b) Produce l'elenco delle stanze libere per ogni tariffa
- c) Nessuna alternativa è corretta
- d) Produce l'elenco delle stanze associate con l'identificatore della tariffa delle prenotazioni effettuate da 'XXX' con data d'inizio compresa fra l'inizio di una tariffa e la sua data di fine

163) La memoria stabile serve al DBMS per memorizzare:

- a) Nessuna alternativa è corretta
- b) Le strutture dati che servono per garantire la persistenza del database
- c) I dati del buffer, prima che vengano riportati nella memoria secondaria
- d) Tutti i record di checkpoint definiti durante il funzionamento

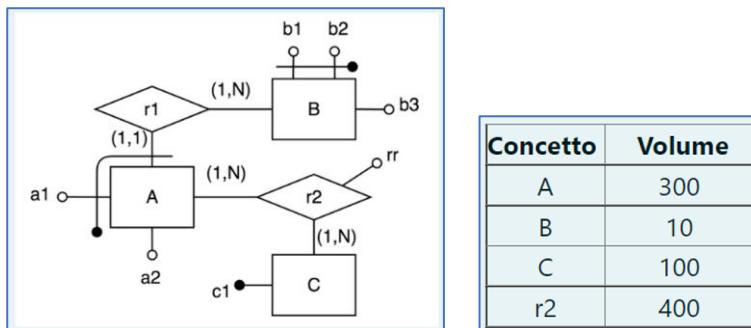
# PRETEST

---

164) L'algoritmo di traduzione di un diagramma concettuale E-R verso il modello logico relazionale ottimizza l'occupazione di memoria per la tabella corrispondente all'associazione R(E1,E2):

- a) Nessuna alternativa è corretta
  - b) Quando la cardinalità di R è 0,1 per E1 o per E2
  - c) Quando la cardinalità di R è 1,1 per E1 o per E2
  - d) Solo quando la cardinalità di R è 1,1 sia per E1 che per E2
- 

165) Dati il diagramma E-R e la tabella dei volumi sottostanti, quanti sono, nel caso peggiore, gli accessi per determinare i valori di **a2** e **b1** relativi a occorrenze di **C** associate a un valore di **rr** pari a 'xxx', qualora i valori diversi di **rr** siano 10, ed equi-probabili?:



- a) 480
  - b) 700
  - c) 440
  - d) Non si può sapere, date le informazioni a disposizione
- 

166) Data l'istanza **T(a,b,c,d,e)** sotto a sinistra, la query alla sua destra:

T				
a	b	c	d	e
a1	b1	4	2021-01-07	3
a1	b2	3	2022-05-20	3
a2	b3	4	2022-03-30	4
a3	b1	1	2021-05-20	(NULL)
a3	b2	2	2022-02-22	2
a4	b3	1	2022-06-01	1
a5	b3	2	2021-05-18	3

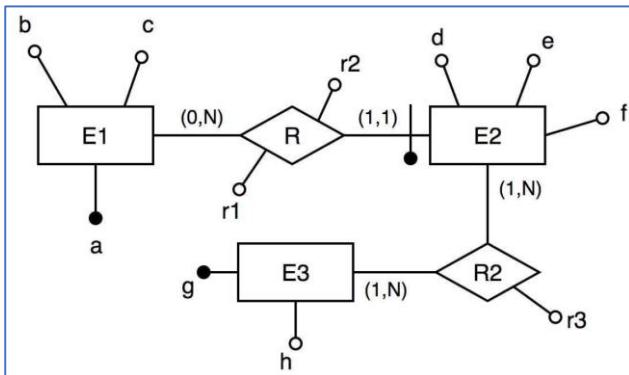
```
SELECT T.a, T.b,
       LAST_VALUE(T.d) OVER(PARTITION BY T.b
                           ORDER BY T.d) as r
FROM T;
```

- a) Per ogni record, restituisce T.a, T.b e la data più recente fra quelle dei record con stesso valore di T.b
  - b) Per ogni valore di T.b, crea una partition con record in ordine crescente rispetto a T.d, e restituisce T.a, T.b e il valore più recente di T.d nella partition
  - c) Nessuna alternativa è corretta
  - d) È errata sintatticamente (non compila)
-

# PRETEST

---

167) Quanti vincoli vengono generati dalla traduzione del seguente diagramma E-R nel modello logico relazionale?:



- a) 3 vincoli di primary key e 4 vincoli di foreign key
  - b) 4 vincoli di primary key e 3 vincoli di foreign key
  - c) Non si può dire osservando il diagramma
  - d) 2 vincoli di primary key e 2 vincoli di foreign key
- 

168) Ogni insieme di dipendenze in forma normale:

- a) Può essere minimizzato
  - b) Può essere minimizzato e ha un solo insieme equivalente
  - c) Nessuna alternativa è corretta
  - d) Non deve essere minimizzato
- 

169) Si consideri la seguente base di dati:

Deputato(codice, nome, cognome, commissione, provincia, collegio)  
 Collegio(provincia, numero, nome)  
 Provincia(sigla, nome, regione)  
 Regione(codice, nome)  
 Commissione(numero, nome, presidente)

Qual è il significato della seguente espressione nel calcolo relazionale dei domini?

{nome:n, cognome:c | Provincia(sigla: sg, nomep: np, regione: cr)  $\wedge$  Deputato(codice:cd, nome:n, cognome:c, commissione:num, provincia: sg, collegio: cg)  $\wedge$  Regione(codice: cr, nome:nr)  $\wedge$  not exists nm, nc . Commissione(numero:nm, nome:nc, presidente:cd)}

- a) Elenca i nomi e i cognomi dei deputati
  - b) Elenca i nomi e i cognomi dei deputati che non sono presidenti di commissione
  - c) Nessuna alternativa è corretta
  - d) Elenca i nomi e i cognomi dei deputati che sono presidenti di commissione
-

# PRETEST

170) Si consideri la seguente base di dati:

```
Deputato(codice, nome, cognome, commissione, provincia, collegio)
Collegio(provincia, numero, nome)
Provincia(sigla, nome, regione)
Regione(codice, nome)
Commissione(numero, nome, presidente)
```

Qual è il significato della seguente espressione nel calcolo relazionale dei domini?

```
{nome:n, cognome:c | Provincia(sigla: sg, nomep: np, regione: cr) ∧ Deputato(codice:cd, nome:n, cognome:c, commissione:num, provincia: sg, collegio: cg) ∧ Regione(codice: cr, nome:nr) ∧ (nr='Lombardia' ∧ not exists nm, nc . Commissione(numero:nm, nome:nc, presidente:cd))}
```

- a) Elenca i nomi e i cognomi dei deputati eletti in Lombardia che non sono presidenti di commissione
- b) Nessuna alternativa è corretta
- c) Elenca i nomi e i cognomi dei deputati eletti in Lombardia
- d) Elenca i nomi e i cognomi dei deputati eletti in Lombardia che sono presidenti di commissione

171) Si consideri il seguente schema di base di dati:

```
Prenotazioni (CodFiscaleCliente, DataChiamata, OraChiamata, IDPrenotazione,
NumOspitiPrenotazione, NumStanzePrenotate, DataInizioPrenotazione,
DataFinePrenotazione)
Cliente(NomeCliente, CognomeCliente, CodFiscaleCliente, RecapitoCliente)
Stanza (NumStanza, CategoriaStanza, NumPianoStanza, NumLettiMaxStanza)
Servizi (NumStanza, TipoStagione, ServiziStanza, CostoStanza)
Stagione (TipoStagione, DataInizioS, DataFineS)
PrenotazioneStanza(IdPrenotazione, NumStanza, DataInizio, DataFine)
```

Qual è il significato dell'espressione algebrica

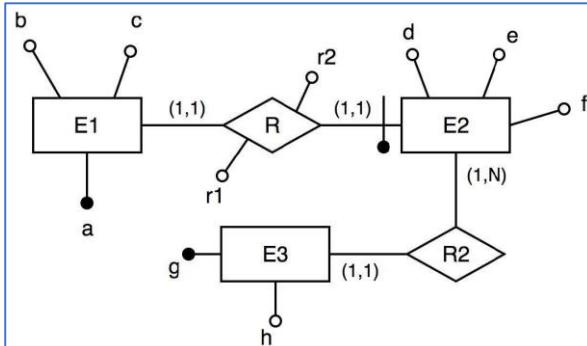
```
πNumStanza( πIdPrenotazione(σCodFiscale='XXX'(Prenotazioni)) ⋈ PrenotazioneStanza)
    ⋈ DataInizio ≥ DataInizioS and DataFine ≤ DataFineS
    π DataInizioS, DataFineS(σTipoStagione='Alta'(Stagione))
)
```

- a) Nessuna alternativa è corretta
- b) Produce l'elenco delle stanze prenotate in alta stagione
- c) Produce l'elenco delle stanze prenotate da 'XXX' in alta stagione
- d) Produce l'elenco delle stanze non prenotate da 'XXX' in alta stagione

# PRETEST

---

172) Dati il diagramma E-R e la tavola dei volumi sottostanti, quanti sono nel caso peggiore, gli accessi per determinare i valori di **g** relativi a occorrenze di **E1** associate a un valore di **r1** pari a ‘xxx’, qualora i valori diversi di **r1** siano 100, ed equi-probabili?:



Concetto	Volume
E1	1000
E3	3000

- a) 3000
  - b) Nessuna alternativa è corretta
  - c) 2130
  - d) 3130
- 

173) Quando possono presentarsi potenziali violazioni dei vincoli di integrità su una base di dati?:

- a) Solo in presenza di operazioni di interrogazione
  - b) Nessuna alternativa è corretta
  - c) Solo in presenza di operazioni di inserimento
  - d) Solo in presenza di operazioni di inserimento e cancellazione
- 

174) Si consideri il seguente insieme di dipendenze **AB**→**C**, **B**→**D**, **B**→**A**, **C**→**E**, **AB**→**E** per la relazione **R**, qual è un insieme minimale equivalente?:

- a) **B**→**C**, **B**→**D**, **B**→**A**, **C**→**E**, **AB**→**E**
  - b) **AB**→**C**, **B**→**D**, **B**→**A**, **C**→**E**
  - c) **AB**→**C**, **B**→**D**, **C**→**E**, **AB**→**E**
  - d) **B**→**C**, **B**→**D**, **B**→**A**, **C**→**E**
- 

175) L'algoritmo di traduzione di un diagramma concettuale E-R verso il modello logico relazionale può non ottimizzare l'occupazione di memoria per la tabella corrispondente all'associazione **R(E1,E2)**:

- a) Quando la cardinalità di **R** è 1,1 sia per **E1** che per **E2**
  - b) Quando la cardinalità di **R** è 1,1 per **E1** o per **E2**
  - c) Quando la cardinalità di **R** è 0,1 per **E1** o per **E2**
  - d) Nessuna alternativa è corretta
-

# PRETEST

---

176) Quando una generalizzazione parziale presente in uno schema E-R è ristrutturata mantenendo sia l'entità madre che le entità figlie:

- a) Nessuna alternativa è corretta
  - b) Si può introdurre una ridondanza, ma ciò è accettabile perché le prestazioni ne risentono positivamente
  - c) Si può introdurre ridondanza e ciò è inaccettabile, ecco perché questa ristrutturazione andrebbe evitata nel caso di generalizzazioni parziali
  - d) Non si introduce mai ridondanza
- 

177) Qual è il caso in cui una query contenente una subquery non può essere riscritta senza usare subquery?:

- a) Quando la subquery è *correlated*
  - b) Non ce ne sono
  - c) Quando la subquery è *noncorrelated*
  - d) Nessuna alternativa è corretta
- 

178) Si consideri la seguente base di dati

```
Deputato(codice, nome, cognome, commissione, provincia, collegio)
Collegio(provincia, numero, nome)
Provincia(sigla, nome, regione)
Regione(codice, nome)
Commissione(numero, nome, presidente)
```

Qual è il significato della seguente espressione nel calcolo relazionale dei domini?:

```
{collegio:cg | Deputato(codice:cd, nome:n, cognome:c, commissione:num,
provincia: sg, collegio: cg)  $\wedge$   $\nexists$  num'. Deputato(codice:cd', nome:n, cognome:c,
commissione:num', provincia: sg, collegio: cg)  $\wedge$  cd  $\neq$  cd'}
```

- a) Elenca collegi dei deputati che hanno omonimi
  - b) Elenca tutti i collegi
  - c) Elenca i collegi in cui non ci sono deputati omonimi
  - d) Nessuna alternativa è corretta
-

# PRETEST

179) Si consideri il seguente schema di base di dati

```
Prenotazioni (CodFiscaleCliente, DataChiamata, OraChiamata, IDPrenotazione, NumOspitiPrenotazione, NumStanzePrenotate, DataInizioPrenotazione, DataFinePrenotazione)
Cliente (NomeCliente, CognomeCliente, CodFiscaleCliente, RecapitoCliente)
Stanza (NumStanza, CategoriaStanza, NumPianoStanza, NumLettiMaxStanza)
Servizi (NumStanza, IDTariffa, ServiziStanza, CostoStanza)
Tariffa (IDTariffa, DataInizioTariffa, DataFineTariffa)
PrenotazioneStanza (IDPrenotazione, NumStanza)
```

Qual è il significato della seguente espressione algebrica:

```

$$\pi_{IDPrenotazione, IDTariffa} \left( \begin{array}{l} \pi_{DataInizioPrenotazione, DataFinePrenotazione, IDPrenotazione} (\sigma_{CodFiscale='XXX'} (Prenotazioni)) \\ \bowtie \\ \pi_{IDPrenotazione} (PrenotazioneStanza) \end{array} \right) \\ \bowtie DataInizioPrenotazione >= DataInizioTariffa \text{ and } DataFinePrenotazione <= DataFineTariffa \\ Tariffa \end{math>$$

```

- a) Nessuna alternativa è corretta
- b) Produce l'elenco delle tariffe associate a prenotazioni effettuate da 'XXX', per periodi interamente contenuti nel periodo di durata di una tariffa
- c) Produce l'elenco delle stanze libere per ogni tariffa
- d) Produce l'elenco delle tariffe di tutte le prenotazioni effettuate da 'XXX'

180) Uno schedule risultate dall'esecuzione delle richieste delle transazioni sulla base del protocollo **2PL** e uno risultante dall'esecuzione delle richieste delle transazioni sulla base del protocollo **2PL** sono view equivalenti:

- a) Talvolta
- b) Sempre
- c) Mai

181) I database NoSQL garantiscono

- a) Nessuna alternativa è corretta
- b) La disponibilità e la scalabilità
- c) La consistenza e la scalabilità
- d) La consistenza e la disponibilità

# PRETEST

---

182) Uno schedule risultante dall'esecuzione delle richieste delle transazioni sulla base del protocollo **2PL stretto** è conflict equivalente ad uno in cui è stato usato il **2PL**

- a) Sempre
  - b) Mai
  - c) Talvolta
- 

183) Si consideri la seguente base di dati

```
Deputato(codice, nome, cognome, commissione, provincia, collegio)
Collegio(provincia, numero, nome)
Provincia(sigla, nome, regione)
Regione(codice, nome)
Commissione(numero, nome, presidente)
```

Qual è il significato della seguente espressione nel calcolo relazionale dei domini?:

```
{nome:nc | Deputato(codice:cd, nome:n, cognome:c, commissione:num, provincia: sg, collegio: cg)  
  \ Commissione(numero:num, nome:nc, presidente:cd) \notin sg', cg', cd' . Deputato(codice:cd', nome:n,  
cognome:c, commissione:num, provincia: sg', collegio: cg') \wedge cd \neq cd'}
```

- a) Elenca i nomi delle commissioni che non hanno un membro omonimo del presidente
  - b) Elenca i nomi delle commissioni che non hanno un presidente
  - c) Elenca i nomi di tutte le commissioni
  - d) Nessuna alternativa è corretta
- 

184) Il result set di una query avente una window function basata su frame fra gli attributi del SELECT:

- a) Non può contenere duplicati, infatti in queste query non serve usare *DISTINCT*
  - b) Non può contenere record con attributi il cui valore è *NULL*
  - c) Nessuna alternativa è corretta
  - d) Contiene solo un record in cui il valore della window function è calcolato considerando un sottoinsieme dei record della partition
- 

185) Si consideri il seguente insieme di dipendenze **AB→CDE, C→ED** per la relazione R, qual è un insieme minimale equivalente?:

- a) **AB→C, AB→E**
  - b) **AB→C, C→ED**
  - c) **AB→C, C→E**
  - d) L'insieme è già minimale
-

# PRETEST

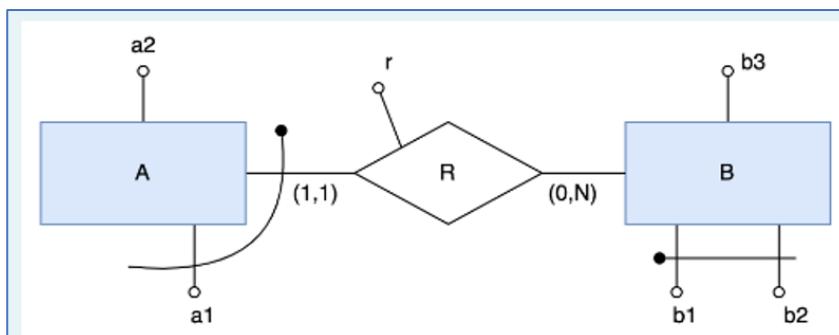
---

186) Si consideri la relazione  $R(A,B,C)$ : il numero  $N(R)$  di tuple di  $R$  è 50000, il numero di valori diversi dell'attributo  $A$  è 50, di  $B$  è 500, e di  $C$  è 1000. Supponendo equiprobabili le distribuzioni dei valori degli attributi, qual è la dimensione del risultato, in termini di numero di tuple, della seguente espressione nel caso peggiore?:

$\pi_C (\sigma_{A='n1' \text{ and } B='n2'}(R))$

- a) 500
  - b) 2000
  - c) Nessuna alternativa è corretta
  - d) 100
- 

187) Dato il seguente diagramma E-R, quanti vincoli sono generati dalla sua traduzione verso il modello logico relazionale?:



- a) 2 vincoli di chiave primaria e 1 vincolo di integrità referenziale
  - b) 3 vincoli di chiave primaria e 1 vincolo di integrità referenziale
  - c) Nessuna alternativa è corretta
  - d) 2 vincoli di chiave primaria e 2 vincoli di integrità referenziale
- 

188) Tra le varie motivazioni per le quali i moderni DBMS relazionali hanno introdotto le stored procedure, una è senz'altro perché:

- a) Le altre alternative non descrivono motivazioni che hanno portato all'introduzione delle stored procedure
  - b) Alcune query sono più facili da risolvere tramite un approccio procedurale
  - c) Un utente possa interagire con un database senza conoscere il linguaggio SQL
  - d) Alcune query, altrimenti, non potevano essere risolte con il solo codice dichiarativo
-

# PRETEST

189) Si consideri il seguente schema di base di dati

```
Prenotazioni (CodFiscaleCliente, DataChiamata, OraChiamata, IDPrenotazione, NumOspitiPrenotazione, NumStanzePrenotate, DataInizioPrenotazione, DataFinePrenotazione)
Cliente(NomeCliente, CognomeCliente, CodFiscaleCliente, RecapitoCliente)
Stanza (NumStanza, CategoriaStanza, NumPianoStanza, NumLettiMaxStanza)
Servizi (NumStanza, IDTariffa, ServiziStanza, CostoStanza)
Tariffa (IDTariffa, DataInizioTariffa, DataFineTariffa)
PrenotazioneStanza(IDPrenotazione, NumStanza)
```

Qual è il significato della seguente espressione algebrica:

$$\begin{aligned} & \pi_{\text{NumStanza}, \text{IDTariffa}}( \\ & ( \\ & \quad \pi_{\text{DataInizioPrenotazione}}(\sigma_{\text{CodFiscale}='XXX'}(\text{Prenotazioni})) \\ & \quad \bowtie \\ & \quad \text{PrenotazioneStanza} \\ & ) \\ & \bowtie \text{ DataInizioPrenotazione} \geq \text{DataInizioTariffa} \text{ and } \text{DataInizioPrenotazione} \leq \text{DataFineTariffa} \\ & \quad \text{Tariffa} \\ & ) \end{aligned}$$

- a) Produce l'elenco delle stanze associate con l'identificatore della tariffa delle prenotazioni effettuate da 'XXX' con data d'inizio compresa fra l'inizio di una tariffa e la sua data di fine
- b) Produce l'elenco delle stanze libere per ogni tariffa
- c) Nessuna alternativa è corretta
- d) Produce l'elenco delle stanze associate con l'identificatore della tariffa di tutte le prenotazioni effettuate da 'XXX', se la prenotazione rientra completamente in una sola tariffa

190) Quando si implementa una materialized view in sync con i raw data, quali devono essere le principali caratteristiche dei trigger di push?:

- a) Nessuna alternativa è corretta
- b) Deve essere veloce e limitare gli accessi ai raw data a quelli basati su clausole *where* (o predicati di join) che usano condizioni sulle chiavi primarie delle tabella raw
- c) Deve essere veloce e ridurre al minimo gli accessi ai raw data
- d) Deve essere veloce ed evitare del tutto gli accessi ai raw data

# PRETEST

---

191) Date le istanze **T** ed **S** sotto, la query successiva restituisce:

T					S	
a	b	c	d	e	a	f
a0	b1	9	2019-05-03	1	a1	6
a1	b1	0	2019-05-18	2	a2	1
a1	b0	(NULL)	2019-05-01	1	a3	1
a3	b0	9	2022-05-18	1		
a3	b2	2	2022-05-18	(NULL)		
a5	b1	1	2022-05-08	2		
a5	b0	9	2022-05-18	1		

```
SELECT *
FROM T NATURAL LEFT OUTER JOIN S
WHERE T.e <> 2;
```

- a) Quattro record
  - b) Tre record
  - c) Cinque record
  - d) Nessuna alternativa è corretta
- 

192) Il result set della query successiva, eseguita sull'istanza **T** sotto:

T					
a	b	c	d	e	
a0	b1	9	2019-05-03	1	
a1	b1	0	2019-05-18	2	
a1	b0	(NULL)	2019-05-01	1	
a3	b0	9	2022-05-18	1	
a3	b2	2	2022-05-18	(NULL)	
a5	b1	1	2022-05-08	2	
a5	b0	9	2022-05-18	1	

```
SELECT T.b, COUNT(*) OVER w AS h
FROM T WHERE T.c < 9
WINDOW w AS(PARTITION BY T.b ORDER BY T.d
            RANGE BETWEEN
            INTERVAL 1 MONTH PRECEDING AND CURRENT ROW);
```

- a) Contiene alcuni record il cui valore di *h* è uguale a 3
  - b) Contiene record con valori di *h* tutti maggiori o uguali di 1
  - c) Contiene record con valori di *h* tutti uguali a 3
  - d) Nessuna alternativa è corretta
-

# PRETEST

---

193) Ogni insieme di dipendenze in terza forma normale:

- a) Può essere minimale
  - b) È necessariamente minimale
  - c) Non può essere ridotto in forma minimale
  - d) Nessuna alternativa è corretta
- 

194) La scrittura dei risultati delle operazioni nella base di dati avviene con maggiore efficienza:

- a) Quando è fatta sempre prima del commit
  - b) Quando può essere fatta sia prima che dopo il commit
  - c) Quando è fatta sempre dopo il commit
  - d) Nessuna alternativa è corretta
- 

195) Data la seguente tabella T, la query a fianco:

T			
a	b	c	d
a1	b2	(NULL)	2
a1	b4	c3	4
a2	b2	c2	5
a1	b1	c1	2
a2	b3	(NULL)	1

```
SELECT T.a, LAST_VALUE(T.d) OVER w AS N  
FROM T  
WINDOW w AS (PARTITION BY T.a ORDER BY T.d DESC);
```

- a) Restituisce due record ('a1',2) e ('a2',1)
  - b) Restituisce lo stesso result set di:  

```
SELECT T.a, MIN(T.d)  
FROM T  
GROUP BY T.a
```
  - c) Nessuna alternativa è corretta
  - d) Per tutti i record con un dato valore di T.a, non restituisce lo stesso valore di T.d
  - e) Per ogni record, restituisce il valore di T.a e l'ultimo valore di T.d, fra tutti i record con lo stesso valore di T.a ordinati in modo decrescente su T.d
  - f) È errata sintatticamente
- 

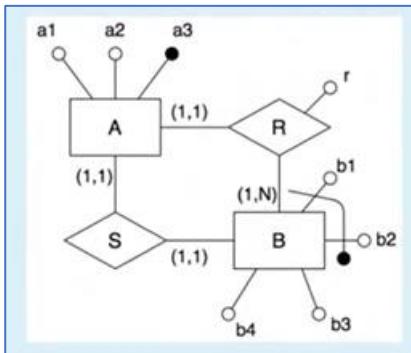
196) Considerando il tipo di linguaggio:

- a) L'algebra relazionale e il calcolo relazionale sono entrambi linguaggi procedurali
  - b) L'algebra relazionale è procedurale, mentre il calcolo relazionale è dichiarativo
  - c) Nessuna alternativa è corretta
  - d) L'algebra relazionale è dichiarativa, mentre il calcolo relazionale è procedurale
  - e) L'algebra relazionale e il calcolo relazionale sono entrambi linguaggi dichiarativi
-

# PRETEST

---

197) Considerato il diagramma E-R in figura:



- a) Non è ridondante
  - b) Non si può dire se sia ridondante o non ridondante
  - c) È errato perché l'entità B, per com'è espressa la sua chiave, deve obbligatoriamente partecipare con cardinalità (1,1) alla relazione R, non (1,N)
  - d) Nessuna alternativa è corretta
  - e) È ridondante
- 

198) Il protocollo 2PL:

- a) Richiede il lock di ogni risorsa immediatamente prima di usarla, non può richiedere il lock con troppo anticipo
  - b) Risolve il problema del deadlock
  - c) Può richiedere il lock su una risorsa solo dopo aver rilasciato il lock su un'altra, è per questo che il 2PL è efficiente
  - d) Nessuna alternativa è corretta
- 

199) La query eseguita sulla tabella tt mostrata alla sua destra:

```

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

tt			
a	b	c	d
a1	b2	(NULL)	2
a1	b4	c3	4
a2	b2	c2	5
a1	b1	c1	3
a2	b3	(NULL)	1

- a) È errata sintatticamente (non compila)
  - b) Restituisce un result set con N=NULL nel record il cui valore dell'attributo b è 'b2'
  - c) Nessuna alternativa è corretta
  - d) Produce un result set senza valori NULL anche se si toglie la clausola WHERE
  - e) Restituisce 4 record
-

# PRETEST

---

200) Indicare la complessità dell'operazione di verificare se uno schema **R** soddisfa la **BCNF**:

- a) Esponenziale
  - b) Costante
  - c) Polinomiale
- 

201) Siano dati i seguenti due schemi di tabelle: **T1(a,b,c,d)** e **T(e,f,g,h,i)**

La query seguente:

```
SELECT DISTINCT(T1.b)
FROM T1
WHERE T1.c EXISTS
(
  SELECT *
  FROM T2
  WHERE T2.i <> T1.d
);
```

- a) È errata sintatticamente
  - b) Calcola il result set della subquery per ogni record di T1 e controlla che T1.c sia fra i valori di T2.g nel result set della subquery
  - c) Restituisce un numero
  - d) Nessuna alternativa è corretta
  - e) Deve avere un result set scalare nella subquery
- 

202) Un trigger **AFTER UPDATE** su una tabella **T**:

- a) Può modificare la tabella T
  - b) Nessuna alternativa è corretta
  - c) È adatto a gestire una business rule
  - d) Serve solo per propagare un aggiornamento eseguito su T in un'altra tabella che contiene informazioni ridondanti (una materialized view oppure una log table)
  - e) Non contiene statement con la keyword NEW perché quando va in esecuzione, il record è già stato aggiornato
- 

203) Un'espressione dipendente dai domini nel calcolo relazionale può essere rappresentata in algebra relazionale:

- a) Sempre
  - b) Mai
  - c) Solo se si è usato il calcolo dei domini
-

# PRETEST

---

204) Data la relazione **R** e i predicati **cond** e **cond1**, la seguente equivalenza in algebra relazionale:

$$\sigma_{\text{cond1}}(\sigma_{\text{cond}} R) \equiv \sigma_{\text{cond}}(\sigma_{\text{cond1}} R)$$

- a) È falsa se ‘cond’ non coinvolge attributi coinvolti in ‘cond1’
  - b) È sempre vera
  - c) È sempre falsa
  - d) È vera solo se ‘cond1’ è un predicato su attributi presenti anche nel predicato ‘cond’
  - e) Nessuna alternativa è corretta
- 

205) Il calcolo dei domini richiede una variabile diversa solo per i domini di nome diverso di ogni relazione:

- a) Vero
  - b) Falso
- 

206) Data una tabella **T(a,b,c,d)**, una tabella pivot:

- a) Nessuna alternativa è corretta
  - b) È una tabella in cui, per ogni attributo t di T, ci sono tanti attributi quanti sono i valori assunti dai record di T sull’attributo t
  - c) Ha, come colonne le righe di T
  - d) Si può omettere scegliendo un attributo t di T, e creando una tabella in cui i valori assunti su t dai record di T divengono i record di tale tabella, assieme a un valore aggregato
- 

207) Il metodo per effettuare il join viene scelto dal DBMS in base:

- a) Alla dimensione delle tabelle coinvolte
  - b) Alla lunghezza delle righe delle tabelle coinvolte
  - c) Alla dimensione delle chiavi delle tabelle coinvolte
- 

208) Si consideri lo schema **R(A,B,C,D,E)** con le dipendenze **A→BC**, **CD→E**, **B→D**, **E→A**, e la decomposizione di **R** in **R1(A,B,C)** e **R2(A,D,E)**. La decomposizione conserva le dipendenze?

- a) Vero
  - b) Falso
-

# PRETEST

---

209) Il risultato **r** restituito dalla stored procedure **p** dichiarata come segue:

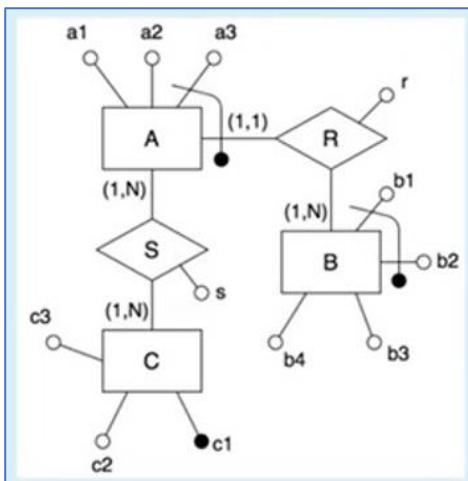
```
DROP PROCEDURE IF EXISTS p;
DELIMITER $$ 
CREATE PROCEDURE p(OUT r INT)
BEGIN
    -- body
END $$ 
DELIMITER ;
```

- a) Può contenere più record, ma di un unico attributo ciascuno
  - b) Nessuna alternativa è corretta
  - c) Potrebbe essere un result set
  - d) È un unico valore
  - e) È sempre un unico record composto da uno o più attributi di tipo intero
- 

210) Il concetto di transazione viene introdotto per risolvere le anomalie di perdita di aggiornamento, di aggiornamento fantasma e di letture inconsistenti

- a) Falso
  - b) Vero
- 

211) Dato il diagramma E-R in figura, quanti vincoli di integrità referenziale introduce?:



- a) 5
  - b) Nessuna alternativa è corretta
  - c) 6
  - d) 4
-

# PRETEST

---

212) Si consideri la seguente base di dati

Deputato(codice, nome, cognome, commissione, provincia, collegio)  
Collegio(provincia, numero, nome)  
Provincia(sigla, nome, regione)  
Regione(codice, nome)  
Commissione(numero, nome, presidente)

Qual è il significato dell'espressione algebrica:

$\pi_{\text{nome,cognome}}(\pi_{\text{presidente}}(\text{Commissione})$   
 $\bowtie_{\text{presidente} \neq \text{codice}}$   
 $\pi_{\text{codice,nome, cognome}}(\text{Deputato}))$

- a) Elencare i deputati delle commissioni
  - b) Elencare i deputati non membri delle commissioni
  - c) Elencare i presidenti di commissione
  - d) Elencare tutti i deputati
- 

213) Il paradigma evento-condizione-azione stabilisce il funzionamento di:

- a) Trigger
  - b) Trigger ed event
  - c) Stored procedure, trigger ed event
  - d) Nessuna alternativa è corretta
- 

214) Considerare una materialized view **M** i cui dati sono basati su quelli di una tabella **T**, sui cui sono effettuate solo operazioni di insert. Se **M** è gestita con incremental refresh in modalità deferred, allora il trigger di push:

- a) Non serve c'è un event che gestisce l'aggiornamento
  - b) Può esserci, ma in quel caso deve essere chiamato da un event
  - c) C'è sempre e, quando scatta, aggiorna la materialized view
  - d) Nessuna alternativa è corretta
- 

215) Le funzioni di aggregazione:

- a) Possono essere usate nel SELECT, nel WHERE e nella HAVING clause
  - b) Possono essere usate solo nella HAVING clause perché esprimono condizioni sui gruppi
  - c) Nessuna alternativa è corretta
  - d) Possono essere usate nel SELECT, ma la query deve contenere un GROUP BY
-

# PRETEST

---

216) Dato il generico record **r** della tabella **T(a,b,c)**, il risultato della window function **lag(c,k)**, dove **k** è un intero positivo, applicata al record **r** di **T**:

- a) Può valere NULL se nella query non c'è la clausola OVER
  - b) Nessuna alternativa è corretta
  - c) Può valere NULL se la query usa la clausola OVER contenente ORDER BY, indipendentemente dalla presenza di PARTITION
  - d) Può valere NULL solo se la query usa la clausola OVER contenente ORDER BY, ma non PARTITION BY
- 

217) Data l'istanza della tabella **M** sotto a destra, la chiamata **CALL proc()**:

```
1 DELIMITER $$  
2 CREATE PROCEDURE proc()  
3 BEGIN  
4     DECLARE dd, ss INTEGER DEFAULT 0;  
5     REPEAT  
6         SET dd = dd+1;  
7         SET ss = ss+dd;  
8     UNTIL dd > (SELECT MAX(M.d) FROM M)  
9     END REPEAT;  
10    SELECT ss;  
11 END $$  
12 DELIMITER ;
```

M				
a	b	c	d	e
a1	b9	c8	32	2021-12-01
a1	b0	c8	30	2022-01-20
a1	b8	c8	32	2021-10-01
a2	b4	c5	30	2021-11-11
a2	b0	c2	30	2021-01-02
a3	b0	c2	32	2022-01-02

- a) Genera un ciclo infinito
  - b) Stampa la somma dei valori diversi dell'attributo *d* dei record di M
  - c) Stampa 32
  - d) Nessuna alternativa è corretta
  - e) Stampa la somma dei numeri minori o uguali a 33
- 

218) Conoscendo lo schema **M(a,b,c,d,e)** la query seguente eseguita su un'istanza generica di **M**:

```
SELECT M.a  
FROM M  
WHERE NOT EXISTS(  
    SELECT *  
    FROM M M2  
    WHERE M2.e <> M.e  
        AND M2.a = M.a  
)
```

- a) Non esegue né la divisione né la differenza insiemistica
  - b) Nessuna alternativa è corretta
  - c) Esegue una differenza insiemistica
  - d) Esegue una divisione insiemistica
-

# PRETEST

---

219) Data l'istanza della tabella **M** sotto a sinistra, i result set delle query **Query 1** e **Query 2** sotto a destra:

M				
a	b	c	d	e
a1	b9	c8	32	2021-12-01
a1	b0	c8	30	2022-01-20
a1	b8	c8	32	2021-10-01
a2	b4	c5	30	2021-11-11
a2	b0	c2	30	2021-01-02
a3	b0	c2	32	2022-01-02

```
1 -- Query 1
2 SELECT MIN(M.e) OVER(ORDER BY M.e)
3 FROM M;
4
5 -- Query 2
6 SELECT MIN(M.e)
7 FROM M
8 ORDER BY M.e;
```

- a) Sono diversi, ma potrebbero essere uguali cambiando l'istanza di M
  - b) Sono uguali e continuano ad esserlo anche se l'istanza di M cambia
  - c) Nessuna alternativa è corretta
  - d) Sono uguali, ma potrebbero essere diversi cambiando l'istanza di M
  - e) Sono diversi e continuano ad esserlo anche se l'istanza di M cambia
- 

220) Si consideri la relazione **R(A,B,C)**: il numero **N(R)** di tuple è **50000**, il numero di valori diversi degli attributi **A** e **B** è **50**, e di **C** è **10000**; ciascun elemento di **A**, **B** e **C** occupa una parola di memoria. Supponendo equiprobabili le distribuzioni dei valori degli attributi, qual è la dimensione in termini di parole occupate dal risultato della seguente espressione nel caso peggiore?:

$\Pi_{C,A} (\sigma_{A='n1' \text{ and } B='n2'}(R))$

- a) 1000
  - b) Nessuna alternativa è corretta
  - c) 2000
  - d) 5
- 

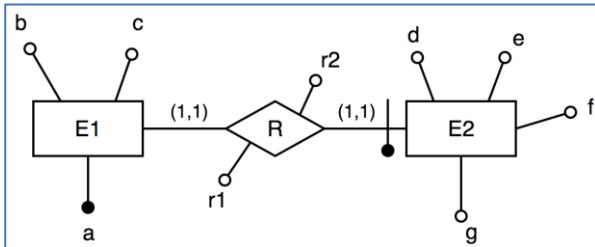
221) In SQL, un vincolo di integrità referenziale sull'attributo **a** viene sempre inserito nella tabella:

- a) Che rappresenta la traduzione di un'associazione tra entità, una della quali ha a come attributo
  - b) Nessuna alternativa è corretta
  - c) Che rappresenta la traduzione di un'associazione tra entità, una della quali usa a come chiave
  - d) In cui a è chiave
-

# PRETEST

---

222) Quando si traduce il diagramma E-R riportato sotto, nel modello logico relazionale:



- a) Nessuna alternativa è corretta
  - b) Si generano due tabelle, E1 ed E2, inserendo r1 ed r2 obbligatoriamente in E2, per la presenza della chiave esterna
  - c) Si generano due tabelle, E1 ed E2, inserendo r1 ed r2 in una di tali tabelle
  - d) E1 ed E2 possono essere tradotte in un'unica tabella oppure in due tabelle, in una delle quali vanno inseriti anche r1 ed r2
- 

223) Una transazione può contenere:

- a) Un commit o un abort su ogni cammino di esecuzione
  - b) Un commit su ogni cammino oppure un abort su ogni cammino
  - c) Nessuna alternativa è corretta
  - d) Un solo commit e un solo abort
- 

224) Si consideri la seguente base di dati

```
Deputato(codice, nome, cognome, commissione, provincia, collegio)
Collegio(provincia, numero, nome)
Provincia(sigla, nome, regione)
Regione(codice, nome)
Commissione(numero, nome, presidente)
```

Qual è il significato dell'espressione algebrica:

```
 $\pi_{\text{nome}, \text{cognome}}(\text{Deputato}) -$ 
 $\pi_{\text{nome}, \text{cognome}}($ 
 $\quad \pi_{\text{numero}}(\sigma_{\text{nome}=\text{'esteri'}}(\text{Commissione}))$ 
 $\quad \bowtie_{\text{numero}=\text{commissione}}$ 
 $\quad \pi_{\text{commissione}, \text{nome}, \text{cognome}}(\sigma_{\text{commissione}=\text{null}}(\text{Deputato}))$ 
 $)$ 
```

- a) Elenca nome e cognome dei deputati della commissione esteri
  - b) Elenca nome e cognome dei deputati che non stanno in nessuna commissione
  - c) Elenca nome e cognome di tutti i deputati
  - d) Nessuna alternativa è corretta
-

# PRETEST

---

225) Si consideri la seguente base di dati

```
Deputato(codice, nome, cognome, commissione, provincia, collegio)
Collegio(provincia, numero, nome)
Provincia(sigla, nome, regione)
Regione(codice, nome)
Commissione(numero, nome, presidente)
```

Qual è il significato della seguente espressione nel calcolo relazionale dei domini?:

```
{nome:np | Provincia(sigla: sg, nome: np, regione: cr) ∧ Deputato(codice:cd, nome:n, cognome:c, commissione:num, provincia: sg, collegio: cg) ∧ Regione(codice: cr, nome:nr) ∧ (nr='Lombardia' ∨ nr≠'Toscana')}
```

- a) Elenca i nomi delle province della Lombardia in cui sono stati eletti deputati
  - b) Nessuna alternativa è corretta
  - c) Elenca i nomi delle province né in Toscana né in Lombardia in cui sono stati eletti deputati
  - d) Elenca i nomi delle province non in Toscana in cui sono stati eletti deputati
- 

226) Lo schedule risultante dall'esecuzione delle richieste delle transazioni sulla base del protocollo **time-stamp** è **view serializzabile**, ma è:

- a) Non view equivalente allo schedule che riproduce l'ordine delle richieste delle transazioni
  - b) Non view equivalente a schedule ottenibili usando il protocollo 2PL
  - c) Nessuna alternativa è corretta
  - d) Non view equivalente a schedule conflict serializzabili
- 

227) Uno schedule risultante dall'esecuzione delle richieste delle transazioni sulla base del **2PL** è **view serializzabile**, ma:

- a) Non è view equivalente a schedule conflict serializzabili
  - b) Non è view equivalente allo schedule che riproduce l'ordine delle richieste delle transazioni
  - c) Nessuna alternativa
  - d) Non è view equivalente a schedule ottenibili usando il protocollo time-stamp
-

# PRETEST

---

228) Quando una transazione abortisce, tutte le sue operazioni vengono disfatte e:

- a) Le operazioni di UNDO vengono solamente inserite nel log, al posto delle operazioni originali
  - b) Le operazioni di UNDO vengono solamente inserite nel buffer
  - c) Le operazioni originali vengono solamente cancellate dal log
  - d) Nessuna alternativa è corretta
- 

229) Una associazione ternaria del modello E-R con cardinalità **(0,1)** con una delle entità coinvolte:

- a) Ogni istanza di quella entità non compare in alcuna terna dell'associazione
  - b) Ogni istanza di quell'entità può partecipare ad al più una terna dell'associazione
  - c) Nessuna alternativa è corretta
  - d) Esistono terne dell'associazione in cui quell'entità ha valore NULL
- 

230) Nella ripresa a caldo e a freddo le informazioni del log vengono utilizzate sempre tutte:

- a) Sì ma in modo diverso
  - b) Nessuna alternativa è corretta
  - c) Sì perché il dump serve a stabilire il before state delle operazioni
  - d) Sì perché nella ripetizione o annullamento delle operazioni si parte dall'ultimo dump
- 

231) L'algoritmo di traduzione di un diagramma concettuale E-R verso il modello logico relazionale può decidere se creare la tabella corrispondente all'associazione **R(E1,E2)**:

- a) In base al fatto che R abbia attributi o no
  - b) In base alla dimensione delle chiavi di E1 e E2
  - c) In base alle cardinalità di R
  - d) Nessuna alternativa è corretta
- 

232) L'istruzione **FETCH**

- a) È eseguita prima che il NOT FOUND handler rilevi la potenziale assenza di record da prelevare
  - b) Non è eseguita se il NOT FOUND handler rileva l'assenza da record da prelevare
  - c) È eseguita solo se il NOT FOUND handler non ha rilevato l'assenza di record da prelevare
  - d) Nessuna alternativa è corretta
-

# PRETEST

---

233) Data l'istanza di T sotto a sinistra, la stored procedure sotto a destra:

T				
a	b	c	d	e
a0	b1	9	2019-04-05	1
a1	b0	NULL	2019-05-01	1
a1	b1	0	2019-04-18	1
a3	b0	3	2022-05-18	1
a3	b2	2	2022-06-18	2
a5	b0	3	2022-07-18	1
a5	b1	1	2022-05-08	2

```
DELIMITER $$  
CREATE PROCEDURE p()  
BEGIN  
    DECLARE e INTEGER DEFAULT 0;  
    DECLARE c CURSOR FOR  
        SELECT T.e INTO e  
        FROM T;  
    scan: LOOP  
        IF e > 1 THEN  
            SIGNAL SQLSTATE '45000'  
            SET MESSAGE_TEXT = 'Errore';  
        END IF;  
    END LOOP scan;  
END $$  
DELIMITER ;
```

- a) Contiene un ciclo (loop) che viene eseguito quattro volte
- b) Nessuna alternativa è corretta
- c) Scorre la tabella T e si ferma, sollevando un errore, non appena trova un record il cui valore di T.e è maggiore di 1
- d) Non compila

---

234) Si consideri la seguente base di dati

```
Deputato(codice, nome, cognome, commissione, provincia, collegio)  
Collegio(provincia, numero, nome)  
Provincia(sigla, nome, regione)  
Regione(codice, nome)  
Commissione(numero, nome, presidente)
```

Qual è il significato delle seguenti espressioni nel calcolo relazionale dei domini?:

```
{provincia: sg, collegio: cg | Deputato(codice:cd, nome:n, cognome:c, commissione:num, provincia: sg, collegio: cg) ∧ Deputato(codice:cd', nome:n, cognome:c, commissione:num', provincia: sg, collegio: cg) ∧ cd ≠ cd' }
```

- a) Elenca provincia e collegio dei deputati che non hanno omonimi
- b) Nessuna alternativa è corretta
- c) Elenca provincia e collegio di tutti i deputati
- d) Elenca provincia e collegio in cui vi sono deputati omonimi

# PRETEST

235) Si considerino le relazioni **R1(A,B)** e **R2(B,C)**: il numero di tuple di **R1** è **50000**, il numero di valori diversi degli attributi **A** e **B** è **500**, e di **C** è **1000**. Supponendo equiprobabile le distribuzioni dei valori degli attributi, qual è la dimensione in termini di tuple del risultato dell'espressione (seguente) nel caso peggiore?:

$\sigma_{A='n1'}(R1) \bowtie \sigma_{B='n2'}(R2)$

- a) 10000
- b) 200
- c) 100
- d) Nessuna alternativa è corretta

236) Si consideri il seguente schema di base di dati

Prenotazioni (CodFiscaleCliente, DataChiamata, OraChiamata, IDPrenotazione, NumOspitiPrenotazione, NumStanzePrenotate, DataInizioPrenotazione, DataFinePrenotazione)  
Cliente(NomeCliente, CognomeCliente, CodFiscaleCliente, RecapitoCliente)  
Stanza (NumStanza, CategoriaStanza, NumPianoStanza, NumLettiMaxStanza)  
Servizi (NumStanza, IDTariffa, ServiziStanza, CostoStanza)  
Tariffa (IDTariffa, DataInizioTariffa, DataFineTariffa)  
PrenotazioneStanza(IDPrenotazione, NumStanza)

Quale è il significato dell'espressione algebrica:

```
 $\pi_{\text{NumStanza}}($   
     $\bowtie$   
     $\sigma_{\text{DataInizioPrenotazione}, \text{DataFinePrenotazione}, \text{IDPrenotazione}}(\sigma_{\text{CodFiscaleCliente}='XXX'}(\text{Prenotazioni}))$   
     $\bowtie$   
     $\text{PrenotazioneStanza}$   
)  
 $\bowtie$  DataInizioPrenotazione >= DataInizioTariffa and DataFinePrenotazione <= DataFineTariffa  
Tariffa  
)
```

- a) Produce l'elenco delle stanze libere per ogni tariffa
- b) Produce l'elenco delle stanze relative alle prenotazioni effettuate da 'XXX', per periodi interamente contenuti nel periodo di una tariffa
- c) Produce l'elenco delle stanze di tutte le prenotazioni effettuate da 'XXX'
- d) Nessuna alternativa è corretta

# PRETEST

---

237) Le dipendenze di un insieme minimale

- a) Hanno a destra un insieme di attributi ciascuno contenuto in una chiave della relazione
  - b) Nessuna alternativa è corretta
  - c) Non hanno né a sinistra una superchiave della relazione né ogni attributo a destra è contenuto in una chiave
  - d) Hanno a sinistra una superchiave della relazione
- 

238) Data l'istanza di T sotto a sinistra, la query sotto a destra:

T				
a	b	c	d	e
a0	b1	9	2019-04-05	1
a1	b0	NULL	2019-05-01	1
a1	b1	0	2019-04-18	1
a3	b0	3	2022-05-18	1
a3	b2	2	2022-06-18	2
a5	b0	3	2022-07-18	1
a5	b1	1	2022-05-08	2

```
SELECT T.c,  
       RANK() OVER(PARTITION BY T.e ORDER BY T.c)  
FROM T  
WHERE MONTH(T.d) > 5;
```

- a) Nessuna alternativa è corretta
  - b) Non restituisce record a pari merito
  - c) Restituisce due record a pari merito
  - d) È errata sintatticamente (non compila)
- 

239) Data l'istanza di T sotto a sinistra, la query a destra:

T				
a	b	c	d	e
a0	b1	9	2019-04-05	1
a1	b0	NULL	2019-05-01	1
a1	b1	0	2019-04-18	1
a3	b0	3	2022-05-18	1
a3	b2	2	2022-06-18	2
a5	b0	3	2022-07-18	1
a5	b1	1	2022-05-08	2

```
SELECT T.a,  
       SUM(T.e) OVER(ORDER BY T.c)  
FROM T  
WHERE T.a <> 'a1';
```

- a) Effettua una somma, quindi restituisce lo stesso risultato anche omettendo "ORDER BY T.c"
  - b) Nessuna alternativa è corretta
  - c) Restituisce un record contenente T.a e la somma dei valori di T.e contenuti in tutti i record aventi quel valore di T.a
  - d) Restituisce un solo record perché non c'è PARTITION BY
-

# PRETEST

---

240) Il deferred refresh necessita di una (o più) log table per memorizzare i dati inseriti/cancellati/modificati fra un refresh e il successivo:

- a) Vero
  - b) Falso
- 

241) I database NoSQL permettono di realizzare tabelle:

- a) Nessuna alternativa è corretta
  - b) Con tuple con elementi tutti dello stesso tipo
  - c) Solo con tuple di dimensione fissa
  - d) Con tuple di dimensione qualsiasi
- 

242) Si consideri la seguente base di dati

```
Deputato(codice, nome, cognome, commissione, provincia, collegio)
Collegio(provincia, numero, nome)
Provincia(sigla, nome, regione)
Regione(codice, nome)
Commissione(numero, nome, presidente)
```

Qual è il significato della seguente espressione nel calcolo relazionale dei domini?:

```
{nome:nc | Deputato(codice:cd, nome:n, cognome:c, commissione:num, provincia: sg, collegio: cg) ∧ n='Mario'
∧ c='Rossi' ∧ Commissione(numero:nm, nome:nc, presidente:cd)}
```

- a) Elenca i nomi dei deputati di commissioni di cui non è presidente Mario Rossi
  - b) Produce il nome della commissione di cui è presidente Mario Rossi
  - c) Nessuna alternativa è corretta
  - d) Elenca i nomi dei deputati della commissione di cui è presidente Mario Rossi
- 

243) I database che scalano facilmente orizzontalmente:

- a) Possono usare server standard
  - b) Nessuna alternativa è corretta
  - c) Hanno bisogno di server molto potenti
  - d) Non hanno bisogno di più server
-

# PRETEST

244) Si considerino le relazioni **R1(A,B)** e **R2(B,C)**: il numero di tuple di **R1** è **50000**, il numero di valori diversi degli attributi **A** e **B** è **500**, e di **C** è **1000**. Supponendo equiprobabile le distribuzioni dei valori degli attributi, qual è la dimensione in termini di tuple del risultato dell'espressione (seguente) nel caso peggiore?:

$\sigma_{A='n1' \text{ and } B='n2'}(R1) \bowtie R2$

- a) 10000
- b) 100
- c) 500
- d) Nessuna alternativa è corretta

245) Si consideri il seguente schema di base di dati

Prenotazioni (CodFiscaleCliente, DataChiamata, OraChiamata, IDPrenotazione, NumOspitiPrenotazione, NumStanzePrenotate, DataInizioPrenotazione, DataFinePrenotazione)  
Cliente(NomeCliente, CognomeCliente, CodFiscaleCliente, RecapitoCliente)  
Stanza (NumStanza, CategoriaStanza, NumPianoStanza, NumLettiMaxStanza)  
Servizi (NumStanza, IDTariffa, ServiziStanza, CostoStanza)  
Tariffa (IDTariffa, DataInizioTariffa, DataFineTariffa)  
PrenotazioneStanza(IDPrenotazione, NumStanza)

Quale è il significato dell'espressione algebrica:

$$\begin{aligned} & \pi_{\text{NumStanza}}( \\ & ( \\ & \quad \pi_{\text{DataInizioPrenotazione}, \text{IDPrenotazione}}(\sigma_{\text{CodFiscaleCliente}='XXX'}(\text{Prenotazioni})) \\ & \quad \bowtie \\ & \quad \text{PrenotazioneStanza} \\ & ) \\ & \bowtie \text{ DataInizioPrenotazione} \geq \text{DataInizioTariffa} \text{ and } \text{DataInizioPrenotazione} \leq \text{DataFineTariffa} \\ & \quad \text{Tariffa} \\ & ) \end{aligned}$$

- a) Produce l'elenco delle stanze delle prenotazioni effettuate da 'XXX' con data d'inizio compresa fra l'inizio di una tariffa e la sua data di fine
- b) Nessuna alternativa è corretta
- c) Produce l'elenco delle stanze di tutte le prenotazioni effettuate da 'XXX', se la prenotazione rientra completamente in una sola tariffa
- d) Produce l'elenco delle stanze libere per ogni tariffa

# PRETEST

---

246) Nel result set di una query con raggruppamento, possono esserci duplicati?:

- a) Sì, perché il raggruppamento può produrre più di un record per gruppo
  - b) Sì, ma solo quando i record del result set contengono attributi non presenti nel predicato di raggruppamento
  - c) No, perché il raggruppamento compatta i record in gruppi, e ciascun record del risultato è quindi diverso
  - d) Nessuna alternativa è corretta
- 

247) Data l'istanza della tabella T sotto a sinistra, la query sotto a destra:

T				
a	b	c	d	e
a0	b1	9	2019-04-05	1
a1	b0	NULL	2019-05-01	1
a1	b1	0	2019-04-18	1
a3	b0	3	2022-05-18	1
a3	b2	2	2022-06-18	2
a5	b0	3	2022-07-18	1
a5	b1	1	2022-05-08	2

```
SELECT T.d
FROM T
WHERE T.c < ALL
(
    SELECT TT.c
    FROM T TT
    WHERE TT.c <> T.c
);
```

- a) Restituisce 2 record
  - b) Restituisce 1 record
  - c) Restituisce un result set vuoto
  - d) Nessuna alternativa è corretta
- 

248) La scelta se usare una **force** o una **flush** viene fatta dal gestore dell'affidabilità in base al carico applicativo?:

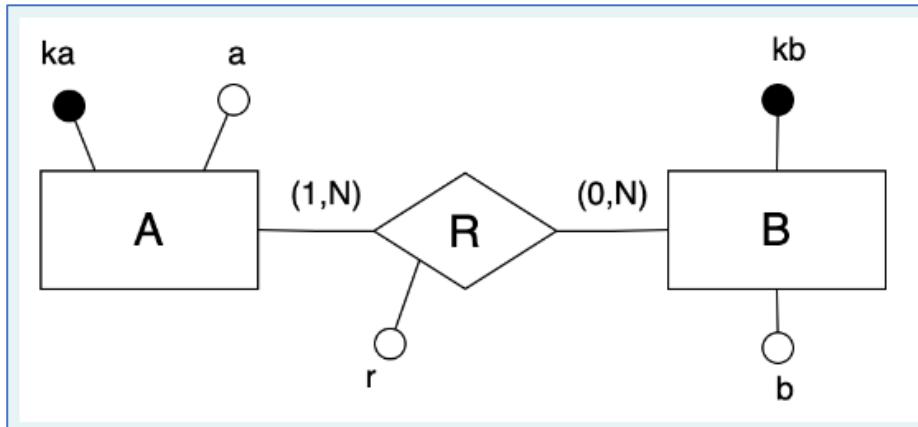
- a) No, ci sono casi in cui il gestore esegue necessariamente una force
  - b) Nessuna alternativa è corretta
  - c) No, in base al carico il gestore sceglie il momento in cui sostituire una force con una flush
  - d) No, in base al carico il gestore sceglie il momento in cui eseguire la force
- 

249) Si consideri il seguente insieme di dipendenze **AB**→**C**, **B**→**D**, **D**→**C**, **C**→**E**, **AB**→**E** per la relazione **R**, qual è un insieme minimale equivalente?

- a) **B**→**D**, **D**→**C**, **C**→**E**
  - b) **AB**→**C**, **C**→**E**, **AB**→**E**
  - c) L'insieme è già minimale
  - d) **AB**→**C**, **B**→**A**, **C**→**E**
-

# PRETEST

250) Considerato il diagramma E-R in figura, contenente un'associazione molti a molti con partecipazione opzionale di B (cardinalità minima 0), quali schemi di tabella produce l'algoritmo di traduzione verso il modello logico relazionale?:



- a) Nessuna alternativa è corretta
- b) A(ka,a,kb,r), B(kb,b)
- c) A(ka,a), B(kb,b), R(ka,kb,r)
- d) A(ka,a), B(kb,b), R(ka,kb,r)

# SOLUZIONI PRETEST

---

## 1) - b

Non è possibile valutare ridondanze senza essere a conoscenza dell'entità di modellazione della base di dati, cioè cosa sta effettivamente memorizzando.

---

## 2) - b

La risposta **a** è sbagliata perché la cardinalità deve essere **0** perché la camera può essere vuota e deve essere **N** perché in una camera ci possono essere più persone → Non **(1,1)**.

La risposta **d** è sbagliata perché **IDdegenza** non fa parte della chiave, è un attributo di associazione.

Le risposte **e**, **f** sono sbagliate perché la correttezza di un diagramma non è data da ciò che avviene dopo nella traduzione, ma è data dal fatto se i costrutti siano usati nella maniera corretta o meno. Inoltre, per entrambe le risposte non va bene nemmeno che “tutti i concetti espressi nella descrizione testuale sono rappresentati” → Il fatto che un paziente può essere ricoverato in stanze diverse, con questo diagramma questo concetto non è rappresentabile. Soluzione **b** perchè:

- Nel testo c'è scritto che “ogni volta un paziente viene ricoverato condivide una stanza”. La relazione **Degenza** verso **PAZIENTE** ha come cardinalità **(1,1)** e quindi un degente non può aver avuto nella sua esperienza più stanze, ma solo una, il che è sbagliato → Andrebbe trasformata l'associazione in entità (dato che ha anche un identificatore **IDdegenza**) e quindi è rappresentabile il fatto che un paziente può essere ricoverato più volte nella stessa stanza oppure in stanze diverse!

La risposta **c** è sbagliata per la soluzione **b**, è proprio la relazione **Degenza** che è sbagliata.

---

## 3) - b

Possiamo trasformare la relazione **R** in tabella quando esistono un numero elevato di tuple di **E1** che non si connettono a **E2**, dato che la partecipazione non è obbligatoria, altrimenti avremmo avuto magari troppi valori **NULL**.

---

## 4) - d

Nella traduzione logica avrei:

- **A(a1, b1, b2, a2) → 1 vincolo di integrità referenziale rappresentato dalla coppia b1,b2 riferente alla tabella B e di conseguenza include anche l'associazione r1**, perché non avrà una tabella separata
- **B(b1, b2, b3) → 0 vincoli di integrità referenziale**
- **C(c1) → 0 vincoli di integrità referenziale**
- **R2(a1, b1, b2, c1, rr) → 2 vincoli di integrità referenziale rappresentati da a1, b1, b2 per la tabella A e c1 per la tabella C**

→ Totale **3 vincoli di integrità referenziale (1+0+0+2=3)**.

# SOLUZIONI PRETEST

---

5) - e

La risposta **b** è sbagliata, a causa della presenza della chiave esterna.

La risposta **c** è sbagliata, perché il vincolo riguarda la coppia di chiavi tra la tabella PAZIENTE e la tabella FIALA VACCINO. Inoltre, si possono vaccinare più pazienti in una DataVaccino a causa della cardinalità (0,N) su PAZIENTE.

La risposta **d** è sbagliata, perché non va ristrutturato un attributo opzionale.

---

6) - e

Nella traduzione dello schema logico dal diagramma E-R gli attributi nella tabella riferita ad A rimangono attributi!

La chiave della tabella A è composta solo dalle chiavi delle due entità E1 ed E2.

---

7) - b

Il diagramma è errato perché sono sbagliate le cardinalità verso le entità AUTOVETTURA e MECCANICO, che dovrebbero essere (0,1) perché un solo meccanico può riparare una sola autovettura, di conseguenza una sola autovettura può essere riparata da un solo tecnico.

---

8) - a

La risposta **a** rappresenta una delle tre possibilità di eliminazione delle generalizzazioni, nella fase di ristrutturazione di uno schema E-R.

---

9) - b

Si utilizzano su R2 gli attributi chiave d1 e d2 perché sulla tabella E2 ci vado sia passando da E3 sia passando da E1, di conseguenza trovo due valori di d.

---

# SOLUZIONI PRETEST

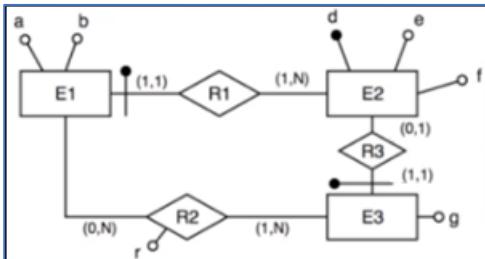
---

## 10) - a

Risposta **d** sbagliata perché la generalizzazione è parziale.

Risposta **b** sbagliata perché non si può parlare di ridondanza senza conoscere la modellazione della base di dati.

Lo schema che si otterrebbe è il seguente. La nuova cardinalità va da (0,1) a (1,1) perché la generalizzazione era parziale, visto che c'è un'occorrenza di E2 che potrebbe non essere di E3, ma quella di E3 è sicuramente occorrenza di E2 (questo rende falsa la risposta **a**):



Nella traduzione logica avrei:

- $E1(a,b,d) \rightarrow 1$  vincolo di integrità referenziale rappresentato da d riferente alla tabella E2, di conseguenza include anche l'associazione R1
- $E2(d,e,f) \rightarrow 0$  vincoli di integrità referenziale
- $E3(g,d) \rightarrow 1$  vincolo di integrità referenziale rappresentato da d riferente alla tabella E2, di conseguenza include anche l'associazione R3
- $R2(d1,d2,g) \rightarrow 2$  vincoli di integrità referenziale rappresentati da:
  - d1 riferente alla tabella E2 passando per la tabella E1
  - d2 riferente alla tabella E2 passando per la tabella E2

→ Si introducono in totale 4 vincoli di integrità referenziale ( $1+0+1+2=4$ ).

---

## 11) - b

Ho usato la chiusura transitiva sugli insiemi di attributi  $\rightarrow (X_i)^{+G} = (X_j)^{+F}$  con gli X<sub>i</sub> che appartengono a F e gli X<sub>j</sub> appartengono ai vari G (presenti tra le risposte).

---

## 12) - b

Risposta ottenuta per esclusione delle altre.

Il metodo sulla chiusura transitiva degli insiemi ha complessità ESPONENZIALE, mentre quello sulla chiusura di insiemi di attributi ha complessità LINEARE (che esclude la risposta **a**). Inoltre, la chiusura sugli insiemi di attributi si fa sulla dipendenza a SX riferente agli attributi della dipendenza a DX, cioè su X<sup>+G</sup> e viceversa (che esclude le risposte **c** e **d**).

---

## 13) - a

Dovrei applicare l'algoritmo di minimizzazione e poi fare la chiusura sull'insieme di attributi, ma posso sorvolare la seconda operazione, dato che basta solo la prima.

---

# SOLUZIONI PRETEST

---

## 14) – RISPOSTA NON PRESENTE (SAREBBERO TUTTE CORRETTE)

Se faccio la chiusura sugli insiemi presenti nelle risposte ho che sono tutte chiavi  
 $\rightarrow E^+ = A^+ = CD^+ = ABCDE$ .

---

## 15) - c

Applico l'algoritmo di minimizzazione.

---

## 16) - d

Applico l'algoritmo di minimizzazione.

---

## 17) - c

Il WHERE esclude l'ultima riga.

Il GROUP BY raggruppa solo sul valore di T1.a=a1.

L'HAVING impone che il conto dei valori di T1.c (senza considerare i duplicati) che sia uguale a 2, quindi su T1.c=c1 e T1.c=c3 → Che sono proprio 2.

Il problema è che non ha senso proiettare su T1.b, dato che non è presente fra gli attributi del GROUP BY e inoltre, l'attributo b è chiave insieme all'attributo a, di conseguenza vengono assunti valori diversi su b per il numero di righe raggruppate → Incoerenza sulla chiave.

---

## 18) - c

Non compila poiché T2.d non può essere proiettato all'esterno della subquery.  
La visibilità dei valori all'interno delle subquery è solo sé stessa o più interna.

---

## 19) - b

L'operatore AVG() in quanto operatore di aggregazione non può essere impiegato all'interno di una direttiva WHERE → Errore di sintassi.

---

## 20) - c

La risposta a è sbagliata perché fa quello che dice, ma per ogni gruppo ne contiene almeno uno di T1.d = NULL, inoltre, il "valore qualsiasi sul restante degli attributi" non è rappresentato dalla query.

La risposta b non è del tutto vera, perché, la nostra query fa un raggruppamento e questa risposta non lo specifica, quindi non conta il GROUP BY.

La risposta d è falsa perché restituisce (a1, 1).

La risposta e è falsa perché non contiene errori.

La risposta f è falsa, perché facendo un raggruppamento almeno un valore è diverso, quello di raggruppamento → In questo caso l'attributo T.a è diverso.

---

# SOLUZIONI PRETEST

## 21) - a

Nel WHERE dovrei confrontare i valori di T1.c con tutti i valori provenienti dalla subquery che sono T1.c (anch'essi) e ciò mi produce un result set vuoto perché il > non è verificato per tutti valori della subquery. Ad un certo punto si troverà un T1.c (query esterna) = T1.c (subquery) che mi farà fallire la condizione del WHERE → Ci vorrebbe al posto di > un >=. La proiezione con il conteggio mi restituisce Q, dato che si conta su un result set vuoto.

## 22) - d

La query fa un''SELF JOIN (ESTERNO SX) di una tabella con sé stessa sui valori diversi di a, e il risultato del JOIN è il seguente (la data per convenienza non è stata inserita su T1 e nemmeno il rimanente degli attributi di T2):

T1				T2	
a	b	c	d	a	c
a1	b1	9	NULL	a2	2
a1	b1	9	NULL	a2	3
a1	b2	1	NULL	a2	2
a1	b2	1	NULL	a2	3
a1	b5	0	NULL	a2	2
a1	b5	0	NULL	a2	3
a2	b1	3	NULL	a1	0
a2	b1	3	NULL	a1	1
a2	b1	3	NULL	a1	9
a2	b3	2	NULL	a1	0
a2	b3	2	NULL	a1	1
a2	b3	2	NULL	a1	9

Diagramma delle relazioni:

- 1° valore di a1 da T1.a con valore diverso su T2.a
- 2° valore di a1 da T1.a con valore diverso su T2.a
- 3° valore di a1 da T1.a con valore diverso su T2.a
- 1° valore di a2 da T1.a con valore diverso su T2.a
- 2° valore di a2 da T1.a con valore diverso su T2.a

Proiettando avrò i 12 record di T1 con il corrispondente T2.a trovato senza restituire nessuna riga con T2.a=NULL, dato che per tutti i record di T1.a c'è almeno un record di T2.a con cui si combina (perchè ogni riga della tabella fa join almeno una volta).

## 23) - b

La suquery è di tipo NON CORRELATED, che non va bene quando abbiamo i costrutti EXISTS/NOT EXISTS → Di conseguenza nella WHERE non ho condizioni valide quindi avrò un result set che conterrà solamente i nomi delle ricette, senza considerare il resto della query.

## 24) - a

Stesso ragionamento della soluzione dell'esercizio 21, solamente che qua non faccio il conteggio, ma proietto solamente, e proietto un result set vuoto!

T1.c (query esterna) non è mai maggiore di T1.c provenienti dalla subquery, dato che una volta per forza T1.c (query esterna) ritroverà per forza sé stesso nel result set della subquery → La WHERE non è mai verificata.

# SOLUZIONI PRETEST

---

## 25) - b

La WHERE mi leva i valori con a<>'a2' → Mi rimangono quelli con a='a1':

H				
a	b	c	d	e
a1	b1	c1	7	2021-01-02
a1	b2	c3	2	2020-12-20
a1	b3	c3	1	NULL
a1	b5	NULL	2	2021-01-31

La classifica r viene partizionata sui valori di a (che ce n'è solo 1, cioè a1) ordinata sui valori di d in maniera crescente così:

- 1 → Relativo a 1
- 2 → Relativo a 2
- 2 → Relativo a 2
- 4 → Relativo a 7

OSS: Ho due valori a pari merito.

La risposta d è sbagliata perché il duplicato su r ce l'ho sul valore 2.

La risposta c è sbagliata perché la query è corretta.

La risposta e è sbagliata perché l'ordinamento ce l'ho su d e non ho mai menzionato nella query il valore c.

La soluzione è b e, anche se il valore 2 è ripetuto, i valori 1, 2, 4 sono diversi fra loro.

---

## 26) - a

---

## 27) - d

La subquery è di tipo CORRELATED, produce un result set vuoto perché la condizione H1.a=H2.a e H1.b=H2.b sono fatte sulla chiave composta da a e b.

Essendo che la chiave è univoca e identifica un unico valore, la condizione H1.e<H2.e non sarà mai verificata perché dovrebbe essere H1.e=H2.e.

La query quindi, compila e produce un result set vuoto.

---

## 28) - e

# SOLUZIONI PRETEST

---

## 29) - c

Per avere un risultato dobbiamo far sì che almeno una volta la CORRELATED subquery torni un result set vuoto.

La prima clausola del WHERE (nella subquery) è sempre vera, perché all'interno della solita tabella c'è sempre un valore di a diverso.

La seconda clausola, anch'essa è vera poiché i valori degli attributi b e c saranno sempre diversi (per i valori in tabella).

Quindi dalla subquery avrò sempre un result set perché c'è almeno un H2.a <> H1.a (ad esempio a1 e a2) e inoltre vale sempre H2.b <> H2.c (perché i valori di questi due attributi sono tutti diversi fra loro) → Quindi il NOT EXISTS non è verificato, e la query mi produrrà un result set vuoto.

---

## 30) - a

Produce errore di tipo semantico, poiché non ho messo a e b nel predicato di raggruppamento.

La risposta d può sembrare giusta, perché collasserebbe tutto in un unico record, sul primo valore dell'attributo raggruppato che verifica la HAVING → Questo accadrebbe se disattivassi l'ONLY FULL GROUP BY.

---

## 31) - a

La WHERE mi mantiene la 4° e 6° riga.

Il GROUP BY separa queste due righe in 2 gruppi distinti, dato che hanno valori distinti sull'attributo d.

Il HAVING non viene verificato perché il conteggio sui valori diversi di a fa esattamente 1 e quindi la query mi produrrà un result set vuoto (che è diverso dal restituire NULL risposta d, anche se in MySQL un result set vuoto è rappresentato proprio da NULL).

---

## 32) - a

Errato sintatticamente, non è possibile fare una query su una tabella che andiamo ad aggiornare.

Rischio LOOP.

# SOLUZIONI PRETEST

## 33) - d

Riportato in seguito l'errore che la seguente procedura dà.

Questo perché la variabile è definita intera e il risultato è una stringa.

The screenshot shows the MySQL Workbench interface. In the SQL editor, the following code is written:

```
1  DELIMITER $$  
2 •  DROP PROCEDURE p;  
3  CREATE PROCEDURE p(OUT ris_ INT)  
4  BEGIN  
5      SET ris_ = (  
6          SELECT DISTINCT c  
7          FROM esame_table  
8      );  
9  END $$  
10 DELIMITER ;  
11 •  CALL p(@a);  
12 •  SELECT @a;
```

In the Output tab, the results of the execution are shown in a table:

Action Output	#	Time	Action	Message	Duration / Fetch
	1	10:02:23	DROP PROCEDURE p; CREATE PROCEDURE p(OUT ris_ INT) BEGIN SET...	0 row(s) affected	0.000 sec
	2	10:02:23	CALL p(@a)	Error Code: 1366. Incorrect integer value: 'c1' for column 'ris_ ' at row 2	0.000 sec

## 34) - b

La query viene eseguita e si modificano i valori nonostante il loop infinito.

La risposta **b** è giusta perché, se il valore è a4 il result set è vuoto e quindi non modifica niente.

## 35) - e

Non compila perché la variabile a non è stata dichiarata precedentemente.

## 36) - a

Non ti sbagliare, UNTIL è una condizione di uscita, non di ripetizione, quindi non va in LOOP infinito, ma esce subito dopo una volta.

@r quindi conterrà il valore 2.

## 37) - d

Restituisce un Errore a tempo di esecuzione in quanto SELECT SUM(T.c)... non produce un singolo risultato, e per questa motivazione non è memorizzabile in una singola variabile.

La SELECT restituisce più di un valore.

## 38) - c

# SOLUZIONI PRETEST

---

39) - d

OVER (PARTITION BY T.a ORDER BY T.d) ← Raggruppa per T.a senza comprimere.

a1 b2 c1 2
a1 b1 c1 3
a1 b3 c3 4

LEAD(T.b,1) ← Acquisisce il valore della sua colonna b, sostituendo con il valore della colonna b della riga successiva.

T.c	N
c1	b1
c1	b3
c3	NULL

---

40) - d

La risposta a è sbagliata perché arriva a 3.

La risposta b è sbagliata perché non effettua la RANK su tutti gli attributi di T.

La risposta c è sbagliata perché compila.

La risposta e è sbagliata perché ha 2 record a pari merito.

---

41) - a

Default Frame:

- Con ORDER BY → Inizio partition → CURRENT ROW
- Senza ORDER BY → Inizio partition → Fine partition

Il LAST VALUE lavora sul frame da inizio a CURRENT ROW; perciò, ogni riga avrà FV=c al suo stesso c.

---

42) - a

La risposta b è sbagliata perché restituisce un valore uguale per ogni riga della stessa partition dato che usa la funzione AVG.

---

43) - e

La risposta a è sbagliata perché AVG è un aggregate function che in presenza di partition lavora su partition, ma se esiste il default frame questa lavorerebbe sul frame; quindi, con l'ORDER BY lavora sul frame (inizio partition - CURRENT ROW).

---

# SOLUZIONI PRETEST

---

## 44) - c

In entrambe le query succede questo: viene sommato il valore della cella precedente alla corrente. Però la funzione lag se siamo alla prima riga ritorna NULL.

a	b	SUM(d) OVER w	a	b	d + LAG(d, 1) OVER ( PARTITION BY a ORDER BY e )
a1	b2	2	a1	b2	NULL
a1	b1	4	a1	b1	4
a1	b3	6	a1	b3	6
a2	b3	1	a2	b3	NULL
a3	b2	1	a3	b2	NULL

---

## 45) - a

L'operatore LAG in quanto prende in questo caso il valore del record precedente rispetto al corrente, produrrà ovviamente come primo record un result set formato da un attributo ed un NULL.

---

## 46) - b

La risposta a è sbagliata perché numera i record, ma a livello di partizione.

La risposta b è giusta perché i duplicati esistono sul result-set, ma non sulla partizione singola.

---

## 47) - d

Grazie alla clausola del WHERE che elimina il terzo record siamo sicuri che ogni partition contiene una sola row.

Quindi la non aggregate DENSE RANK non genera mai ovviamente ex-aequo.

---

## 48) - b

La risposta d inganna, dato che una partition con due row che ha lo stesso valore sull'attributo d, restituirà lo stesso valore anche nella seconda.

---

## 49) - a

La risposta d trae in inganno, dato che il valore del record precedente viene preso all'interno della partizione creata.

La soluzione è a perché ha senso utilizzarla quando si ha un ORDER BY.

---

# SOLUZIONI PRETEST

---

50) - b

Le risposte a e d sono sbagliate, perché fanno la proiezione finale su numero, che non è un attributo richiesto, in quanto l'esercizio dice di voler "nome e cognome".

La proiezione su codice riferente alla tabella Deputato sembrerebbe superflua, ma è utile per non avere casi di omonimia.

---

51) - d

La chiave di relazione del JOIN serve per rappresentare i presidenti della commissione sull'uguaglianza presidente=codice.

---

52) - a

Per avere lo stesso numero di tuple di R bisognerebbe proiettare la chiave della tabella (su B e C), che non è presente in nessuna delle proiezioni, delle soluzioni proposte.

---

53) - c

Elencare tutte le tuple che non hanno X equivale a dire:

→ Tutte le tuple – tutte le tuple che hanno X.

In questo caso, ricette che non contengono tabasco = tutte le ricette – le ricette che contengono tabasco.

---

54) - a

La query trova il nome delle ricette, per le quali il codice ingrediente non sta tra (suqbquery) i codici degli ingredienti che hanno nome dell'ingrediente diverso dallo zenzero → Quindi trova il nome delle ricette che hanno nome dell'ingrediente uguale allo zenzero.

---

55) - a

La risposta d può trarre in inganno, ma per essere soddisfatta abbiamo bisogno dell'operatore sottrazione, mentre nella risposta esatta, i deputati possono anche essere parte della commissione bilancio, basta che facciano parte anche di una commissione diversa da questa.

---

56) - d

La proiezione esterna su nome è riferita alla tabella Provincia.

---

# SOLUZIONI PRETEST

---

57) - c

La prima espressione: "Nome e cognome dei deputati che fanno parte di una commissione"

- (meno)

La seconda espressione: "Nome e cognome dei deputati che fanno parte di una commissione, diversa dalla commissione esteri"

→ Ottengo la risposta c.

Per la prima espressione l'appartenenza del Deputato alla Commissione è dovuta al !=NULL.

Per la seconda espressione il fatto che la Commissione del Deputato sia diversa dalla Commissione=esteri è dovuta al != nel JOIN.

---

58) - a

Il fatto che ci si riferisca al 'presidente della commissione' nella soluzione, si ha perché viene fatto il JOIN su presidente:pr (della tabella Commissione) con codice:pr (della tabella Deputato).

---

59) - b

---

60) - b

L'espressione algebrica non ha senso, perché nella prima proiezione su NomeRicetta non viene specificata la tabella da cui viene presa, e quindi una sottrazione insensata.

---

61) - a

La prima espressione:

- Prendo gli A e B della tabella R e ci sottraggo gli A e B della tabella R che fanno JOIN con la tabella S sulla condizione C=D → Mi rimangono gli A e B non della tabella S

La seconda espressione:

- Prendo gli A e B per cui non esistono c ed e tali che viene fatto il JOIN tra le tabelle R ed S su C:c=D:e → Mi rimangono gli A e B non della tabella S
- 

62) - c

Il JOIN verrebbe fatto (anche se c'è il diverso) su numero:num (della tabella Commissione) con commissione:num' (della tabella Deputato).

---

# SOLUZIONI PRETEST

---

## 63) - a

Tra le risposte la dicitura: “... non della Toscana né della Sicilia” è rappresentato dalla condizione (nr != ‘Sicilia’ AND nr != ‘Toscana’) → Quindi, considero regioni diverse dalla Sicilia e dalla Toscana **CONTEMPORANAMENTE**.

È presente, inoltre, un JOIN tra sigla:sg (della tabella Provincia) e provincia:sg (della tabella Deputato) → Questo implica che c’è almeno un deputato (eletto come nella risposta a).

---

## 64) - a

La prima espressione:

- Prendo gli A e B della tabella R e ci sottraggo gli A e B della tabella R che fanno JOIN con la tabella S sulla condizione C=D → Mi rimangono gli A e B non della tabella S

La seconda espressione:

- Prendo gli A e B della tabella R e non esiste e tale che viene fatto il JOIN tra le tabelle R ed S su C:c=D:c → Mi rimangono gli A e B non della tabella S
- 

## 65) - a

Non è possibile usare l’operatore di unione U nel calcolo delle tuple.

---

## 66) - d

La ridenominazione la faccio esternamente prima di I perché altrimenti non potrei fare il JOIN all’interno tra le tabelle R1 e R2 sugli attributi B:d e D:d.

---

## 67) - c

È una caratteristica fondamentale del DBMS relazionale sulle operazioni.

---

## 68) - b

Caratteristica delle transazioni. Proprietà ACID (al posto di persistenza, potrei avere durata).

---

## 69) - d

Per definizione.

---

## 70) - d

È una tabella temporanea che mantiene risultati utili all’interno di una sessione, cancellate alla fine della sessione, la risposta a può trarre in inganno, ma è falsa perché un utente con il comando DROP TABLE può distruggerla.

---

# SOLUZIONI PRETEST

---

## 71) - c

La risposta **a** è sbagliata perché può essere eseguita tramite una stored procedure senza problemi, usando una politica on demand.

La risposta **b** è sbagliata perché non copia mai il contenuto della log table, ma lo rielabora.

La risposta **d** è sbagliata perché non può essere di tipo immediate.

La **MV** può essere aggiornata tramite un incremental refresh, esso può avvenire in 3 modalità:

1. Partial refresh → Trasferimento parziale del log
  2. Complete refresh → Trasferimento totale del log
  3. Rebuild → Ricalcolo completo dalle row table (full refresh)
- 

## 72) - e

La risposta **a** è sbagliata perché quelle sono le politiche della materialized view, non dell'incremental refresh.

La risposta **b** è sbagliata perché non copia il contenuto, il log contiene solo i dati necessari e che non si trovano ancora nella materialized, ma non effettua una copia precisa, è solo un modo per trasferire le informazioni in modo più veloce, essendo di una dimensione molto minore, e soprattutto come dice la risposta **d** può anche copiarne una parte.

La risposta **c** è sbagliata perché possiamo per esempio eseguirlo anche tramite event.

La risposta **d** è sbagliata per lo stesso motivo della risposta **b**.

---

## 73) - d

In un'architettura distribuita (come da testo) di server ce ne possono essere più di uno.

Inoltre, il server necessita di un'interfaccia (ciò nega la risposta **a**) e ha un ruolo passivo perché esegue le richieste dei client (ciò nega la risposta **b**).

---

## 74) - c

Solitamente i sistemi **NoSQL** sacrificano l'affidabilità rispetto ai database relazionali

Inoltre, i database relazionali non scalano orizzontalmente, quindi non si ampliano su più server.

---

## 75) - b

Non c'è una regola di base, si effettua sempre la ripresa a freddo nel momento in cui abbiamo un guasto hardware, indipendentemente da cosa viene fatto prima o dopo dalle transazioni.

Al momento del guasto si effettua la ripresa a freddo, ripetendo i procedimenti sugli oggetti danneggiati, per rimanere in uno stato consistente.

---

## 76) - d

# SOLUZIONI PRETEST

---

77) - b

*La risposta a è sbagliata, perché è visibile all'interno della sessione dell'utente.*

*La risposta d è sbagliata, perché viene vista come una variabile globale.*

---

78) - a

*La risposta b è sbagliata perché se raggruppiamo per la chiave, il numero resta invariato, non c'entra usare o meno l'operatore di aggregazione.*

*La risposta d è sbagliata perché può contenere nel GROUP BY attributi e poi non proiettarli.*

*La soluzione è a perché è l'opposto della risposta d.*

*OSS: In un raggruppamento l'importante è che gli attributi presenti nella SELECT vadano a finire nel GROUP BY, a meno che non si raggruppi sulla chiave primaria.*

---

79) - c

*Vale per la proprietà di isolamento (da ACID).*

---

80) - b

*Per i sistemi NoSQL, l'assenza di uno schema consente l'uso di record con un numero di attributi variabile.*

---

81) - c

*La risposta a e la d sono sbagliate, perché la serializzabilità si verifica durante l'esecuzione di uno schedule, non dopo quando è ottenuto.*

*La risposta b è sbagliata, perché possono lavorare anche sugli stessi oggetti.*

---

82) - c

*VSR impone view-equivalenza di uno schedule ad uno seriale, mentre non sempre vale il viceversa → Di conseguenza nella soluzione va bene ‘possono’.*

---

83) - c

*Due espressioni per essere equivalenti, devono produrre lo stesso risultato, qualunque sia l'istanza in termini di righe e colonne, quindi lo stesso schema.*

---

84) - a

*Il gestore dell'affidabilità garantisce la lettera A (affidabilità) e la lettera D (durabilità) della parola ACID → Dopo che una transazione fa commit, il suo risultato deve rimanere permanente nel tempo.*

---

# SOLUZIONI PRETEST

---

85) - c

E è l'insieme dato inizialmente.

---

86) - c

Per la 3FN il miglior algoritmo deterministico conosciuto, nel caso peggiore ha complessità esponenziale.

Per ogni dipendenza di R si guarda se a SX c'è una chiave o superchiave, e per ognuna di esse si vanno a guardare a DX tutti i possibili sottoinsiemi delle chiavi.

---

87) - a

---

88) - a

---

89) - b

---

90) - b

L'indice secondario deve per forza contenere riferimenti a tutti i valori della chiave, i quali devono essere sempre raggiungibili, e per questo è denso.

---

91) - d

La risposta a è errata perché 2PL implica la VSR.

La risposta b è errata perché 2PL implica la CSR.

La risposta c è errata perché alcuni schedule 2PL possono essere eseguiti anche con il TS.  
VEDERE GRAFICO TECNICHE DI GESTIONE CONCORRENZA!

---

92) - b

La risposta a è errata perché non hanno mai TS uguali.

La risposta c è errata perché possono avere operazioni di scrittura, sennò tutte sarebbero concorrenti.

La risposta d è errata per lo stesso motivo di quell'altra, sarebbero tutte concorrenti in qualsiasi situazione con questa premessa.

---

93) - b

Il metodo dichiarativo è l'effettiva semantica SQL e implica il calcolo relazionale basato su ennuple (tuple).

---

# SOLUZIONI PRETEST

---

**94) - b**

Una transazione non espone i suoi stati intermedi per mezzo della I di isolamento (da ACID).

---

**95) - a**

Nel calcolo relazionale si possono esprimere espressioni ‘senza senso’ → Dipendenti dal dominio degli attributi.

Nell’algebra relazionale invece hanno senso → Indipendenti dal dominio degli attributi.

---

**96) - a**

Essendo i sistemi NoSQL basati su architetture distribuite, per essi vale il TEOREMA DI BREWER-CAP, che ci riporta che questi sistemi possono essere o meno consistenti.

---

**97) - d**

Il DBMS si basa sulla gestione delle transazioni.

---

**98) - c**

La risposta **a** è falsa perché può essere eseguito anche con molti altri protocolli tipo TS.

La risposta **b** è falsa perché 2PL è un sottoinsieme del CSR.

VEDERE GRAFICO TECNICHE DI GESTIONE CONCORRENZA!

---

**99) - c**

Il fatto che ci si possa riferire alla stessa tabella, lo si può notare con un vincolo di integrità referenziale verso una relationship ricorsiva.

---

**100) - c**

La force nel checkpoint:

- Scrive in memoria di massa, le pagine del buffer modificate dalle transazioni che hanno fatto commit (quindi rende definitivi i risultati)
- 

**101) - b**

Guardare risposta esercizio precedente!

---

**102) - c**

S2PL implica 2PL, il quale a sua volta implica VSR.

---

# SOLUZIONI PRETEST

---

103) - d

Le risposte a, b e c sono sbagliate perché una temporary table ha tempo di vita della sessione.

---

104) - b

La risposta c è sbagliata perché non si chiama tramite la CALL, quella si usa per le stored procedure.

La risposta d è sbagliata perché da definizione la stored function restituisce solo un valore.

---

105) - a

Sempre, vengono scritti prima i record nel log rispetto a quelli sulla memoria secondaria.

---

106) - b

Sarebbe vera se Y fosse insieme della chiave, oppure se fosse la chiave stessa!

---

107) - e

La risposta a è sbagliata perché non sappiamo neanche il significato di una relazione, quindi non sappiamo se perderemo informazione.

La risposta b è sbagliata perché non sappiamo l'interpretazione del diagramma.

La risposta c è sbagliata perché ci sono due associazioni che vanno su due entità diverse e non compiono alcun ciclo, su queste entità.

La risposta e è esatta perché non ha cardinalità (1,1) su una chiave esterna della relazione R1 verso la tabella E1, verso la quale la frequenza deve essere obbligatoria.

---

108) - b

Per esempio, partendo dal checkpoint e salendo a ritroso fino al begin-transaction, possiamo trovare alcune transazioni che non hanno fatto commit → Quindi, queste vanno a finire nell'insieme UNDO, che rappresenta le transazioni da disfare.

---

109) - c

La normalizzazione si può usare durante la progettazione concettuale, sulla verifica di ridondanze, partizionamenti di entità/relazioni, ma non è obbligatoria.

---

110) - b

Potrebbe essere vuoto l'insieme del REDO, poiché ci possono non essere transazioni che hanno fatto commit.

---

# SOLUZIONI PRETEST

---

111) - b

Se abbiamo già i dati necessari nel buffer per eseguire l'operazione, allora non dobbiamo andare a prenderli in memoria secondaria.

---

112) - a

La risposta a è per definizione.

OSS: Il modello concettuale non considera e non fa distinzione fra i vari modelli logici.

---

113) - d

Il protocollo S2PL rilascia il lock solo dopo il commit, quindi non c'è il problema delle letture sporche.

---

114) - e

La risposta a è sbagliata perché usa sempre il log.

La risposta b è sbagliata perché può essere complete.

La risposta c è sbagliata perché se si esegue complete allora la log table si svuota.

La risposta d è sbagliata perché non è necessario fissare un numero massimo di row su cui opera il refresh.

---

115) - c

La risposta a è sbagliata perché ne esiste solo uno di schedule seriale equivalente (soluzione c), quello in cui ci sono le transazioni che rispettano l'ordine di BEGIN.

La risposta d è sbagliata perché lo schedule è eseguibile allora esiste uno schedule seriale equivalente → Il TS come il 2PL se viene applicato produce un risultato, che fa sì che lo schedule venga accettato → Il risultato dice che lo schedule è conflict-serializzabile e quindi esiste uno schedule seriale corrispondente, ed è unico.

---

116) - b

Nel calcolo delle tuple non è possibile usare l'operatore insiemistico d'unione U.

---

117) - a

L'equivalenza algebrica impone che il risultato delle espressioni sia lo stesso, ma i passaggi potrebbero essere diversi e bisogna valutare i risultati intermedi perché possono far variare i costi dell'operazione.

---

118) - c

# SOLUZIONI PRETEST

---

## 119) - b

Nell'architettura three-tier è presente un server applicativo che separa il client dal server.

---

## 120) - a

Per la risposta **c** la ‘partecipazione opzionale’ dice che la cardinalità minima sia 0 e nel tradurre la relazione(associazione) tra le due entità in tabella, implica che la tabella abbia solamente le coppie di E1 ed E2 che esistono → La tabella, quindi, non può avere valori nulli! La risposta **b** è sbagliata, perché potrebbe essere utile anche se non coincidono, dato che non sappiamo il significato di entrambe le entità.

La risposta **d** è sbagliata, perché si può sempre tradurre con una tabella.

La risposta **e** è sbagliata perché non ha per nulla senso.

La soluzione è la risposta **a** perché posso valutare cosa mi conviene, cioè:

- Se il numero di valori nulli che metto accorpando le associazioni da una delle due parti è basso rispetto alla quantità dei dati → Le tengo accorpate
  - Se il numero di valori nulli che metto accorpando le associazioni da una delle due parti è alto rispetto alla quantità dei dati → Mi conviene fare la tabella separata mettendo le coppie esistenti
- 

## 121) - b

La soluzione **b** è ottenuta per esclusione delle altre.

Partiamo dal fatto che non sappiamo su cosa si raggruppa.

La risposta **c** è falsa perché, se proietto su un sottoinsieme del predicato di raggruppamento potrei avere duplicati → Quindi usare il DISTINCT ha senso.

La risposta **d** è falsa perché, siccome si raggruppa sulla chiave formata da a,b non ci sono duplicati per l'univocità della chiave → Quindi il DISTINCT è inutile.

La risposta **e** è falsa perché, per esempio se raggruppo sulla chiave e poi proietto sugli aggregati, vado a vedere su un sottoinsieme degli attributi di raggruppamento (cioè la chiave) e quindi, potrei avere duplicati → Quindi il DISTINCT è utile.

La risposta **a** è falsa perché, è utile anche quando è presente la chiave:

- Per esempio, se raggruppo sulla chiave e poi proietto valori aggregati, posso andare ad avere valori duplicati → Quindi il DISTINCT è utile anche quando tra gli attributi di raggruppamento ho la chiave
- 

## 122) - a

I database relazionali scalano verticalmente (approccio su server centralizzato).

I database NoSQL scalano orizzontalmente (approccio su server distribuito).

---

## 123) - b

Le letture sporche si verificano quando una transazione abortisce.

---

# SOLUZIONI PRETEST

---

124) - d

---

125) - b

La soluzione **b** è ottenuta escludendo sia la **a** che la **c**.

La dicitura “al momento del recovery in entrambi i casi” implica che devo considerare:

- “...aspettare i commit di tutte le transazioni iniziate” → Non c’è da rifare nulla; quindi, non ho la lista REDO
  - “...può rifiutare nuovi commit” → Situazione ‘classica’ dove ho qualcosa da disfare (lista UNDO) e qualcosa da rifare (lista REDO)
- 

126) - c

Generalmente è sparso, ma può essere anche denso.

---

127) - b

La scrittura si fa sia prima sia dopo il commit della transazione.

---

128) - c

Il 1° è un NATURAL JOIN ed avrà solo 3 attributi (quelli in comune, A e B presi una sola volta). Il 2° è un EQUI-JOIN ed avrà 5 attributi (quelli in comune, A e B ripetuti).

---

129) - b

La 3NF è sempre raggiungibile e, si possono ottenere sempre decomposizioni in questo tipo di forma normale senza perdite, poiché conserva sia le dipendenze che i JOIN.

---

130) - d

L’utente è la persona che gestisce la base di dati (andrebbe chiesto al professore per essere più sicuri), e questo può cancellare la procedura tramite il comando → DROP PROCEDURE.

---

131) - a

Ci può essere un controllo effettivo sul trigger stesso, che ci fa uscire subito senza fare altro.

---

132) - d

Percorrendo il log a ritroso dal CK fino al DUMP, gli insiemi di UNDO e REDO possono riempirsi, e di conseguenza si dovrebbero fare e disfare le operazioni.

---

# SOLUZIONI PRETEST

---

133) - a

Per definizione.

---

134) - c

Il fenomeno si previene attraverso l'utilizzo del S2PL, che non è tra le risposte ma in generale, se avviene, bisogna riportare la base di dati al momento precedente allo schedule che ha causato problemi, e questo viene effettuato tramite un abort in cascata delle transazioni, tutte quelle che compongono lo schedule.

---

135) - d

Da non confondere con il fatto che una transazione che fa commit è tale, solo quando il record di log viene scritto.

---

136) - e

---

137) - b

Per definizione.

---

138) - c

Un vincolo può intersecare più tabelle (legate tra loro tramite politica di vincolo) e un inserimento/modifica su una tabella, di conseguenza comprometterebbe anche le altre  
→ Politica CASCADE.

---

139) - b

Data la dimensione del blocco Lb e quella del record Lr, la parte intera del loro rapporto è:  
→ |Lb/Lr|.

---

140) - c

Per risultati intermedi si intendono quelli interni.

Per R1 il risultato intermedio è il seguente:

$$G_A = \chi(R)$$

che, come risultato, ci dà 50000/50=1000 per R1.

Per R2 invece, il risultato intermedio è il seguente:

$$G_C = \ln^1(R)$$

che, come risultato, ci dà 50000/100=500 per R2.

---

# SOLUZIONI PRETEST

---

## 141) - a

La selezione interna mi toglie le righe e mi dà  $50000/100=500$  record, quelli rimasti.

La proiezione esterna mi toglie le colonne, in questo caso la colonna C e non influisce sulla dimensione del risultato.

Le tuple rimaste che rappresentano la dimensione, sono sempre 500 (per entrambe le colonne A e B).

---

## 142) - c

La selezione interna mi toglie le righe e mi dà  $50000/10000=5$  record, quelli rimasti.

La proiezione esterna mi toglie le colonne, in questo caso la colonna C e non influisce sulla dimensione del risultato.

Le tuple rimaste che rappresentano la dimensione, sono sempre 5 (per entrambe le colonne A e B).

---

## 143) - a

La selezione interna mi toglie le righe e mi dà  $50000/50=1000$  record, quelli rimasti.

La proiezione esterna mi toglie le colonne, in questo caso le colonne A e B e non influisce sulla dimensione del risultato.

Le tuple rimaste che rappresentano la dimensione, sono sempre 1000 (per la colonna C).

---

## 144) - c

La proiezione interna mi toglie la colonna C e mi dà  $100*100=10000$  che sarà il nuovo numero di tuple totale su cui ci andrà poi fatta la selezione.

La selezione esterna mi toglie le righe e mi dà  $10000/100=100$  record, che saranno le tuple rimaste, che rappresentano la dimensione del risultato.

OSS: Se avessi avuto una selezione esterna su C avrei avuto 0 come risultato, visto che la colonna è scomparsa.

---

## 145) – b

Per definizione.

# SOLUZIONI PRETEST

---

## 146) - b

Prendo nome e cognome dentro la relazione Deputato.

Di tutti i deputati non li considero tutti ma quelli per cui:

- Non esiste una tupla in Commissione, dove c'è un qualunque numero (nm) e nome (nc) di Commissione, il cui presidente ha un codice (cd) uguale al codice di Deputato (cd), questo mi fa fare il JOIN e mi rappresenta il fatto che il deputato è un presidente
- I deputati (presidenti) si guardano uno per uno, se il loro codice compare in quelli di Commissione → Non si prendono.

Riassumendo: prendo i nomi dei i cognomi dei deputati, per i quali non esistono commissioni, delle quali loro sono presidenti.

---

## 147) - c

La soluzione c è ottenuta per esclusione delle altre.

La risposta b è sbagliata e lo posso rappresentare con due casi:

1. Se la WHERE prima di un GROUP BY toglie tutti i record, i gruppi non esistono più
2. Se la clausola HAVING dopo il GROUP BY anch'essa toglie tutti i record, i gruppi anche qua non esistono più

La risposta d è sbagliata perché la clausola WHERE elimina i record NON i gruppi, i quali se fosse necessario, verrebbero eliminati dalla HAVING.

La risposta a è sbagliata perché la risposta è posta al contrario → Nel SELECT ci può essere un sottoinsieme degli attributi che sono nel GROUP BY, non il contrario!

---

## 148) - a

La funzione prende T, laddove T.e>1 e questo pezzo elimina il record che ha T.e=NULL e quello che ha T.e=1.

I record rimanenti si ordinano partizionando su T.e → Quindi sui valori 2,3,4 rimasti su T.e

- Per il valore 2 ho 1 record solo
- Per il valore 3 ho 3 record
- Per il valore 4 ho 1 record solo

L'ordinamento poi si fa in base a T.c:

- Per i record con T.e=2 e T.e=4 non posso parlare a priori di pari merito perché ho un record soltanto per questi due valori
- Per i record con T.e=3 ho i valori 4,3,2 di T.c → Anche qui non ho pari merito

→ La soluzione è che restituisce una classifica senza pari merito.

ATTENZIONE: la risposta b si esclude perché se cambiamo l'istanza potremmo trovarci anche dei pari merito nelle partition.

OSS: Quando si parla di "PARI MERITO (EX-AEQUO)" ci si riferisce sempre alle partition!

---

# SOLUZIONI PRETEST

---

## 149) – b

L'evento recurring scatta con periodicità tramite la dicitura "ON SCHEDULE EVERY..." .

→ Non è obbligatorio avere STARTS e ENDS perché ad esempio se mandiamo *in esecuzione* (dopo la compilazione) l'evento, dopo una periodicità P se non metto ENDS, l'evento può andare avanti per sempre.

---

## 150) – d

La risposta d viene dall'esclusione delle altre risposte e potrebbe essere rappresentata da una possibile risposta "qualche volta" → Questo perché TS è incluso in VSR, di conseguenza qualche TS può essere VSR.

La risposta c è sbagliata, perché TS e 2PL sono incomparabili, sull'equivalenza.

La risposta a potrebbe essere giusta ma è incompleta perché: uno schedule VSR può essere equivalente ad uno TS ma deve essere anche CSR.

VEDERE GRAFICO TECNICHE DI GESTIONE CONCORRENZA!

---

## 151) – b

Viene fatto un prodotto cartesiano a causa di un CROSS JOIN perché manca la proiezione di IDPrenotazione su PrenotazioneStanza e quindi, non trovando l'attributo a comune (IDPrenotazione) di JOIN con la tabella Prenotazioni combina tutto (tutte le stanze vengono date a tutte le prenotazioni).

OSS: CodFiscale sulla selezione nell'espressione algebrica non è coerente con CodFiscaleCliente nelle tabelle Prenotazioni e Cliente, in sede d'esame farselo specificare prima di dare una soluzione, perché ciò porta ad un CROSS JOIN.

---

## 152) – a

Da INSERT...fino a GROUP BY:

- Inserisce su M, andando in T, raggruppando su A e per ogni record di A, ci mette la somma di C

→ Con queste operazioni ho fatto il BUILD della Materialized View.

Da CREATE...fino a WHERE:

- Il Trigger mi dice "dopo che si è un inserito un record in T, aggiorna M"
- Mette il valore di tot a tot con sommato, il nuovo valore di C (che c'è nel record appena inserito prima nella BUILD), laddove la riga M.a è uguale al nuovo valore inserito nella BUILD (NEW.a)

Considerando le risposte la d è **sbagliata** perché non ha errori di sintassi.

Le risposte b e c dicono praticamente la stessa cosa (mantenere *in sync* equivale all'implementare l'*immediate refresh*).

→ Entrambe sono **sbagliate** perché non consideriamo che in T si possa inserire T.a con un valore diverso da quelli che sono presenti (cioè diverso da a1, a2, a3, a4, a5) → Il che mi porta a far scattare il trigger senza fare un vero aggiornamento.

---

# SOLUZIONI PRETEST

---

153) – d

Come risposta verrebbe da dire la c ma è errata l'espressione algebrica perché:

- Il JOIN tra i deputati viene fatto su n, c, num, sg, cg e imponendo che non esistano num, sg, cg risulta incompatibile con il JOIN → Avrei dovuto mettere per la seconda tabella Deputato gli attributi così: num', sg', cg', e di conseguenza anche dopo il not exists
- 

154) – b

Faccio 1000 accessi su A per controllare che a1 valga 'xxx' o 'yyy' → Mi chiede il caso peggiore e quindi devo vederli tutti.

Siccome non posso sapere a quanti sono collegati devo fare la scansione completa su R1 quindi 4000 accessi e da questo si recupera b2 e kc passando per B che mi collega sulla tabella C, tramite la chiave.

Per sapere il numero di valori di c a cui è associato devo fare la scansione completa su C che mi dà 50 accessi.

→ In totale avrò 1000+4000+50=5050 accessi.

---

155) – a

Dato che ho l'AND considero le selezioni separatamente.

Inizio verificando la selezione, su quella più selettiva (in questo caso è *indifferente*, inizio con A).

A='n1' su R data da → 50000/500=100 elementi.

Su questi 100 elementi ci vado a vedere la parte dell'AND cioè B='n2'; quindi, se ho elementi A='n1' con B='n2'.

È possibile che i B siano tutti uguali a 'n2' quando ho A='n1' con 100 elementi? Si, al massimo sono 100 elementi anch'essi.

Potrei buttare via qualcosa ma mi serve il caso peggiore, cioè 100 elementi.

Facendo la proiezione di A non ho nulla di nuovo ma con quella di C sì.

→ Considero che i C sono 1000 elementi e possono essere tutti diversi da A ma, per come A è stato ottenuto dalla selezione, decido di non buttare via nulla, e mi tengo una tabella di C come su A, cioè C di 100 elementi.

Per la proiezione di A e C ho che entrambi occupano una parola e quindi in totale sono 2 parole e 2 proiezioni di 100 elementi → 100\*2=200 elementi.

# SOLUZIONI PRETEST

---

## 156) - c

Insieme minimale → Dispone di una regola di ottimizzazione sintattica: non ci sono né lettere né dipendenze che si possono togliere.

La normalità → Mi dice invece, che a SX di una dipendenza ci deve essere una superchiave oppure a DX ogni elemento è contenuto in una chiave.

→ Se prendo un insieme in forma normale e lo minimizzo (insieme minimale) rimane in forma normale, quindi vale la risposta c!

---

## 157) - c

T.a <> 'a3' → Mi da due record da **non** considerare perché T.a=a3, e sono:

a3	b1	1	2021-05-20	(NULL)
a3	b2	2	2022-02-22	2

T.b <> 'b1' → Mi da due record da **non** considerare perché T.b=b1, e sono:

a1	b1	4	2021-01-07	3
a3	b1	1	2021-05-20	(NULL)

L'OR mi fa escludere il record con T.a <> 'a3', oppure quello con T.b <> 'b1', oppure con entrambi; mi accorgo però che per la seguente tupla l'OR non è verificato:

a3	b1	1	2021-05-20	(NULL)
----	----	---	------------	--------

La non verifica è dovuta al fatto che T.a='a3' e T.b='b1' → Quindi non può verificarsi che abbia T.a <> 'a3', oppure che abbia T.b <> 'b1' → Rendono l'OR falso, quindi viene tolto solo questo record.

---

## 158) - b

Applico l'algoritmo di minimizzazione.

La minimizzazione riguarda le parti SX quando c'è più di 1 termine.

La parti SX si portano in forma canonica.

Inoltre, si possono togliere dipendenze, nel caso non servissero:

- Parto considerando AB→C e controllo se ho A che determina B oppure se ho B che determina A  
In questo caso ho B→A cioè, B che determina A → Quindi in AB→C non mi serve A e mi rimane la dipendenza B→C
- Considero adesso B→A, e me lo tengo per il discorso precedente
- Considero C→E, e me lo tengo perché non ho altri percorsi per arrivare ad E, fino a questo punto.
- Considero AB→E → Questa dipendenza la posso togliere perché è superflua, dato che per arrivare ad E ci vado da B→C a C→E, che è un cammino più lungo ed è quello che mi serve, nella minimizzazione

# SOLUZIONI PRETEST

---

## 159) – b

Uno *schedule view-serializzabile* può essere *conflict-serializzabile*, ma uno *view-equivalente* ad uno *schedule 2PL* deve essere *conflict-serializzabile*!

Questo perché il protocollo *2PL* restituisce sempre uno *schedule conflict-serializzabile*.  
VEDERE GRAFICO TECNICHE DI GESTIONE CONCORRENZA!

---

## 160) – a

La selezione interna è composta da due selezioni:

1. Selezione su C='n1' data da 50000/1000=50 record rimasti dalla prima selezione
2. Selezione su C='n2' data da 50000/1000=50 record rimasti dalla seconda selezione

L'OR mi impone di prenderle entrambe → Quindi 50+50=100 tuple rimaste in totale dal risultato intermedio.

La proiezione esterna mi toglie le colonne, in questo caso la colonna C e non influisce sulla dimensione del risultato.

Le tuple rimaste, che rappresentano la dimensione, sono sempre 100 (per entrambe le colonne A e B).

---

## 161) – d

“Si, ma non sempre” lo posso spiegare con un caso particolare:

- Se considero di restituire più parametri in uscita → Con la STORED FUNCTION non posso farlo, perché questa ha il vincolo di restituire 1 valore scalare (funzionano similmente alle funzioni di aggregazione)

I parametri di OUT di una STORED PROCEDURE invece, possono essere in numero >1.

---

# SOLUZIONI PRETEST

---

162) – d

Considerando che CodFiscale (sulla selezione) sia CodFiscaleCliente della tabella Prenotazioni (motivo spiegato nell'osservazione finale) si applica la successiva spiegazione.  
SPIEGAZIONE espressione algebrica:

- Si prendono le prenotazioni fatte da una persona il cui codice fiscale è 'XXX'

$\sigma_{\text{CodFiscale}='XXX'}$  (Prenotazioni)

si combinano con le prenotazioni delle stanze tramite il seguente JOIN



PrenotazioneStanza

e se ne prendono la data di inizio prenotazione e l'id di prenotazione  
(mi serve per il JOIN)

$\pi_{\text{DataInizioPrenotazione}, \text{IdPrenotazione}}$

- Di questo risultato precedente si prende la data di inizio prenotazione  
(mi serve per il JOIN) e si mette in combinazione con il seguente risultato

- Si confronta la data di inizio della prenotazione con la data di inizio della tariffa e la data della fine tariffa

$\bowtie_{\text{DataInizioPrenotazione} \geq \text{DataInizioTariffa} \text{ and } \text{DataInizioPrenotazione} \leq \text{DataFineTariffa}}$   
Tariffa

- Alla fine, l'espressione algebrica produce una coppia fatta dal numero della stanza e la tariffa per quella stanza

$\pi_{\text{NumStanza}, \text{IDTariffa}}$

→ Si ottiene quindi per ogni prenotazione fatta da 'XXX' per una determinata stanza, la tariffa relativa alla data d'inizio della prenotazione (che è la risposta d).

La risposta a è sbagliata perché non sappiamo se la prenotazione ci rientra completamente, visto che ne sappiamo solo l'inizio e non la fine.

La risposta b non ha senso e la risposta c è sbagliata.

OSS: CodFiscale sulla selezione nell'espressione algebrica non è coerente con CodFiscaleCliente nelle tabelle Prenotazioni e Cliente, in sede d'esame farselo specificare prima di dare una soluzione, perché ciò porta ad un CROSS JOIN.

---

163) – b

La memoria stabile è una memoria che non può danneggiarsi, ed è un posto dove le informazioni contenute non possono essere perse.

In memoria stabile si mette il LOG → Completo da tutte le strutture come i CK e i DUMP, perché altrimenti non riusciremmo a garantire i recovery dai vari FAULT che si presentano.

La risposta c non va bene perché solitamente i dati del buffer vengono persi periodicamente, quando si presentano guasti software e, di conseguenza vanno poi recuperati.

La risposta d no perché nella memoria stabile ci deve stare tutto il LOG, compreso dei DUMP e non solamente dei CK.

---

# SOLUZIONI PRETEST

---

## 164) – c

Per ‘ottimizzazione’ si intende minor occupazione di memoria.

In questo caso non si valuta una tabella specifica ma un caso generico.

L'unica soluzione è la risposta **c**, in quanto è possibile sempre avere un'ottimizzazione in questo caso, perché si può andare ad eliminare la relazione corrispondente alla relazione scelta.

Per esempio, se abbiamo R con cardinalità (1,1) verso E1, possiamo eliminarla e inglobarla in E1.

La risposta **b** è sbagliata a prescindere, perché non è un'ottimizzazione che si fa, anzi inserendo una delle due relazioni si ottimizza, quindi improbabile togliere.

---

## 165) – c

Essendo rr un attributo di r2 → Gli elementi di r2 vanno visti tutti, quindi 400 elementi.

Di questi 400 elementi, metto da parte quegli rr che hanno il valore che mi interessa ‘xxx’ e, visto che ho i valori diversi di rr pari a 10 devo fare → 400/10 = 40 elementi riguardanti xxx. In r2 ho le chiavi di A (a1,b1,b2) e di C (c1).

Per quei 40 elementi che ho trovato, che corrispondono a rr, mi prendo il relativo b1 che ho già tra le chiavi presenti in r2.

Invece, a2 non ce l'ho, quindi devo accedere tramite r2 su A con la chiave di A (a1,b1,b2), per trovare il valore corrispondente di a2 → Usare r2 implica l'utilizzo di 400 elementi che, sommati ai 40 messi da parte precedentemente → Fa sì che io ottenga 440 elementi.

---

## 166) – c

La LAST\_VALUE (come la FIRST\_VALUE) sono Window Functions” e lavorano su FRAME, non su PARTITION.

Spiegazione della query:

- Prendiamo il valore di T.a, il valore di T.b e l'ultimo valore di T.d, su una partizione fatta per T.d e ordinata per T.b → Avremo una partizione per b1, b2, b3, ordinati in maniera ascendente sulle date!
- La query per ogni partizione, fatta per T.b, va a prendere la data più recente (l'ultima data)

Il problema della query è che ‘è scritta male’, perché se si utilizzano la LAST\_VALUE senza la definizione di FRAME e usiamo la clausola ORDER BY → Utilizzeremo il DEFAULT FRAME che ci farà sempre fermare alla CURRENT ROW (senza andare dove vogliamo).

→ Quindi non ci restituisce l'ultimo valore della partizione ma semplicemente affianca al T.d di una certa riga il suo T.d per causa del DEFAULT FRAME.

La risposta **b** è sbagliata, perché non ci restituisce il valore più recente nella partition.

La **a** e la **d** anch'esse sono sbagliate.

---

# SOLUZIONI PRETEST

---

167) – b

Premessa sulla tabella R che viene inglobata dall'entità E2, con i suoi attributi.

I vincoli di chiave primaria sono 4 e sono:

1. La chiave di E1 formata da a
2. La chiave di E2 formata da d, e, f, a
3. La chiave di E3 formata da g
4. La chiave della relazione R2 formata da g, d, e, f, a

I vincoli di foreign key (integrità referenziale) sono 3 e sono:

1. Sulla relazione R ho 2 vincoli di integrità referenziale, 1 rappresentato da d, e, f, a per la tabella E2 e 1 rappresentato da g per la tabella E3
  2. Sull'entità E2 ho 1 vincolo di integrità referenziale rappresentato dalla chiave esterna a riferente alla tabella E1
- 

168) – a

---

169) – b

Ai fini delle risposte dell'esercizio, le tabelle Provincia e Regione insieme ai loro attributi non servono.

Il JOIN tra le tabelle Regione e Provincia è fatto da codice:cr con regione:cr e, tra le tabelle Provincia e Deputato è fatto da sigla:sg con provincia:sg.

Fasi dell'espressione:

- Considera il nome e cognome dei deputati tali che  

nome:n, cognome:c | Provincia(sigla: sg, nomep: np, regione: cr)  $\wedge$  Deputato(codice:cd, nome:n, cognome:c, commissione:num, provincia: sg, collegio: cg)  $\wedge$  Regione(codice: cr, nome:nr)
- Non esiste un numero e un nome di commissione, per la quale quei deputati sono presidenti  

not exists nm, nc . Commissione(numero:nm, nome:nc, presidente:cd)

→ Quindi non considero i deputati presidenti di commissione  
(Il fatto che intenda i deputati siano presidenti di commissione, è rappresentato dal JOIN tra Deputato e Commissione, fatto da codice:cd con presidente:cd)

# SOLUZIONI PRETEST

170) – a

Il JOIN tra le tabelle Regione e Provincia è fatto da codice:cr con regione:cr e, tra le tabelle Provincia e Deputato è fatto da sigla:sg con provincia:sg.

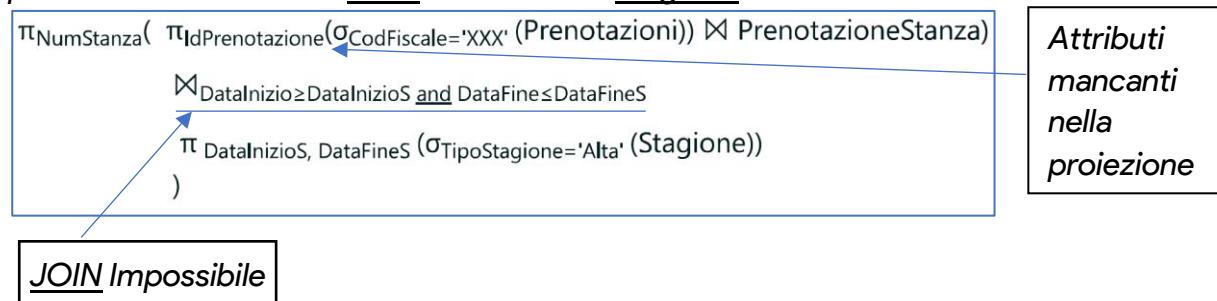
Fasi dell'espressione:

- Considera il nome e cognome dei deputati tali che  
$$\text{nome:n, cognome:c} \mid \text{Provincia(sigla: sg, nomep: np, regione: cr)} \wedge \text{Deputato(codice:cd, nome:n, cognome:c, commissione:num, provincia: sg, collegio: cg)} \wedge \text{Regione(codice: cr, nome:nr)}$$
- Appartengono alla Lombardia  
$$\text{nr='Lombardia'}$$
- E non esiste un numero e un nome di commissione, per la quale quei deputati sono presidenti  
$$\text{not exists nm, nc . Commissione(numero:nm, nome:nc, presidente:cd)}$$

→ Quindi non considero i deputati presidenti di commissione  
(Il fatto che intenda i deputati siano presidenti di commissione, è rappresentato dal JOIN tra Deputato e Commissione, fatto da codice:cd con presidente:cd)

171) – a

Considerando che CodFiscale (sulla selezione) sia CodFiscaleCliente della tabella Prenotazioni (motivo spiegato nell'osservazione finale) si applica la successiva spiegazione. Viene fatto un prodotto cartesiano a causa di un CROSS JOIN perché manca la proiezione su DataInizioPrenotazione e DataFinePrenotazione su Prenotazioni e quindi, non trovando poi l'attributo a comune JOIN con la tabella Stagione combina tutto.



OSS: CodFiscale sulla selezione nell'espressione algebrica non è coerente con CodFiscaleCliente nelle tabelle Prenotazioni e Cliente, in sede d'esame farselo specificare prima di dare una soluzione, perché ciò porta ad un CROSS JOIN.

172) – b

# SOLUZIONI PRETEST

---

173) – b

*Soluzione scelta per esclusione delle altre.*

*Si producono violazioni da una tabella interna verso una tabella esterna:*

- *Per la tabella interna → A seguito di operazioni di modifica*
  - *Per la tabella esterna → A seguito di operazioni di modifica/cancellazione*
- 

174) – d

*Si applica l'algoritmo di minimizzazione.*

---

175) – c

*Quando una delle due entità, o entrambe hanno cardinalità (1,1) → Questo comporta l'eliminazione di una tabella aggiuntiva per R, quindi c'è risparmio (ottimizzazione) di memoria, il che esclude la risposta a e la risposta b.*

*Quando invece, ho la cardinalità (0,1) verso una delle due entità E1/E2 e gli faccio fare comunque l'eliminazione della tabella per R accorpandola in una delle due entità E1/E2 → Potrebbe non ottimizzarsi la memoria, perché ci sarebbero valori nulli, sull'entità dove abbiamo accorpato la relazione R (soluzione a).*

*Andrebbero valutati quanti sono i valori nulli, rispetto alla dimensione della tabella eventuale per R, in cui ci sono solo le coppie che esistono, cioè le chiavi delle due entità che rappresentano un legame vero.*

---

176) – d

*Una pseudo-ridondanza può essere rappresentabile dal fatto che gli elementi che stanno nel padre, stanno anche nei figli, però essendo separati vuol dire che → Ci sono degli attributi diversi che caratterizzano il padre rispetto agli attributi che caratterizzano i figli, ma ciò non è una vera ridondanza!*

*“...andrebbe evitata nel caso di generalizzazioni parziali” dalla risposta c → No perché se viene tolta ad esempio l'entità padre, riportando tutti sui figli, potrei avere degli elementi che non interessano i figli!*

*Questo tipo di ristrutturazione data dal testo dell'esercizio, è una delle tre possibili ristrutturazioni sulle generalizzazioni.*

---

177) – b

*Ogni query con subquery può essere riscritta con una versione JOIN equivalente!*

---

# SOLUZIONI PRETEST

---

178) – c

Fasi dell'espressione:

- Considera i collegi tali che  
 $\text{collegio:cg}$
- Prendi i deputati che stanno in quei collegi (di una determinata provincia)  
 $\text{Deputato}(\text{codice:cd}, \text{nome:n}, \text{cognome:c}, \text{commissione:num}, \text{provincia: sg}, \text{collegio: cg})$
- Ed escludi i deputati di quegli stessi collegi (di quella stessa provincia) con lo stesso nome e cognome → JOIN su provincia:sg, collegio:cg, nome:n, cognome:c per entrambe le tabelle Deputato  
 $\text{Deputato}(\text{codice:cd}, \text{nome:n}, \text{cognome:c}, \text{commissione:num}, \text{provincia: sg}, \text{collegio: cg}) \wedge \# \text{num}' . \text{Deputato}(\text{codice:cd'}, \text{nome:n}, \text{cognome:c}, \text{commissione:num'}, \text{provincia: sg}, \text{collegio: cg})$
- Che hanno codice diverso, cioè sono deputati diversi (per non far riferimento allo stesso deputato)  
 $\text{cd} \neq \text{cd'}$

Riassumendo: si prendono i collegi dei deputati, per i quali non esistono commissioni, di deputati con lo stesso nome e cognome, dello stesso collegio di provincia, con codice diverso.

---

179) – b / a

È stata scelta anche la risposta b per l'osservazione successiva.

OSS: CodFiscale sulla selezione nell'espressione algebrica non è coerente con CodFiscaleCliente nelle tabelle Prenotazioni e Cliente, in sede d'esame farselo specificare prima di dare una soluzione, perché ciò porta ad un CROSS JOIN.

---

180) – a

Essendo 2 schedule eseguibili tramite protocollo 2PL, possono essere view-equivalenti tra loro → Solo se riguardano le stesse transazioni, altrimenti non lo sono → Quindi 'talvolta'.

---

181) – b

Per il Teorema CAP (Brewer) andrebbe bene anche la risposta c, solo che la consistenza è garantita 'prima o poi' e quindi, si preferisce avere la disponibilità (per come viene intesa dalla professorella)

---

# SOLUZIONI PRETEST

---

## 182) – c

Se prendo uno schedule in cui eseguo le richieste delle transazioni in qualche ordine, usando il S2PL e un altro in cui uso il 2PL, ed entrambi funzionano.

→ Tutti e due sono conflict-serializzabili, ma in generale non sono conflict-equivalenti.  
Potrebbero essere ma non sempre!

---

## 183) – a

Fasi dell'espressione:

- Si prendono i nomi delle commissioni tali che  
**nome:nc**
- Si prendono i deputati che sono presidenti di commissione → JOIN tra codice:cd della tabella Deputato e presidente:cd della tabella Commissione  
Deputato(codice:cd, nome:n, cognome:c, commissione:num, provincia: sg, collegio: cg)  
^ Commissione(numero:num, nome:nc, presidente:cd)
- Per questi deputati si prendono solo quelli dove non esiste un altro deputato con lo stesso nome e lo stesso cognome, della stessa commissione → JOIN su nome:n, cognome:c, commissione:num per entrambe le tabelle Deputato  
Deputato(codice:cd, nome:n, cognome:c, commissione:num, provincia: sg, collegio: cg)  
^ Commissione(numero:num, nome:nc, presidente:cd)  
cognome:c, commissione:num, provincia: sg', collegio: cg')
- Che hanno codice diverso, cioè sono deputati diversi (per non far riferimento allo stesso deputato)  
**cd ≠ cd'**

---

## 184) – c

Per esclusione delle altre.

La risposta **d** no perché ad esempio, quando si utilizza la funzione LAST\_VALUE(), dipendentemente dall'uso dell' ORDER BY noi possiamo considerare tutta la partition.

La risposta **a** no perché ci possono essere i duplicati, e quindi il DISTINCT si può mettere e inoltre, in alcuni casi ci deve andare obbligatoriamente.

La risposta **b** no perché se consideriamo funzioni basate su frame che guardano un range di date, quando vi è presente un ordinamento sopra, usando la funzione LAG se non si trova nulla prima vengono posti dei valori NULL.

---

## 185) – b

Applico l'algoritmo di minimizzazione.

---

# SOLUZIONI PRETEST

---

## 186) – d

Dato che ho l'AND considero le selezioni separatamente.

Inizio verificando la selezione, su quella più selettiva.

B='n2' su R data da → 50000/500=100 elementi.

Su questi 100 elementi ci vado a vedere la parte dell'AND cioè A='n1' quindi se ho elementi A='n1' con B='n2'.

La selezione su A mi darà 1000 elementi (dati da 50000/50).

È possibile che i valori di A siano tutti uguali a 'n1' quando ho B='n2' con 100 elementi? Si, al massimo sono 100 elementi anch'essi.

Gli elementi che voglio li ottengo facendo la proiezione su c (dove ci sono tutti valori diversi).

È possibile che le 100 tuple considerate dalla selezione, abbiano lo stesso valore di a, lo stesso valore di b, e tutti i valori diversi di c → Il risultato rimane quindi di 100 elementi!

---

## 187) – a

I vincoli di chiave primaria sono 2 e sono:

- La chiave di A formata da a1,b1,b2
- La chiave di B formata da b1,b2

Di vincolo di integrità referenziale ce n'è solo 1 ed è la chiave esterna b1,b2 riferente alla tabella B.

---

## 188) – c

Le stored procedure garantiscono che un DBMS possa dotarsi di funzionalità precompilate all'interno del server e richiamabili con una CALL senza sapere il linguaggio SQL.

Genericamente tra i vari server di rete, transita proprio la CALL per alleggerirne il carico.

---

## 189) – c

A prescindere dall'osservazione successiva, qua viene fatto un prodotto cartesiano (CROSS JOIN), poiché è stato proiettato l'attributo DataInizioPrenotazione dalla tabella Prenotazioni, che non è sufficiente per il JOIN tra le tabelle Prenotazioni e PrenotazioneStanza → Viene combinata la data di inizio prenotazione con tutte le stanze

$\Pi_{DataInizioPrenotazione}(\sigma_{CodFiscale='XXX'}(Prenotazioni))$   
⊗  
PrenotazioneStanza

OSS: CodFiscale nella selezione nell'espressione algebrica non è coerente con CodFiscaleCliente nelle tabelle Prenotazioni e Cliente, in sede d'esame farselo specificare prima di dare una soluzione, perché ciò porta ad un CROSS JOIN.

---

# SOLUZIONI PRETEST

---

190) – a

Quando una materialized view è *in sync* con i raw data, non c'è il trigger di push → Questo c'è solamente quando si parla di *incremental refresh*.

---

191) – a

Il NATURAL (JOIN) viene fatto sull'attributo omonimo a (senza ripetizioni) e per il LEFT OUTER JOIN vengono mantenuti i record della tabella di SX (cioè T) che soddisfano il JOIN, e viene messo NULL alla tabella di DX (cioè S) ai record che non fanno JOIN ottenendo il seguente risultato:

T				S	
a	b	c	d	e	f
a0	b1	9	2019-05-03	1	(NULL)
a1	b1	0	2019-05-18	2	6
a1	b0	(NULL)	2019-05-01	1	6
a3	b0	9	2022-05-18	1	1
a3	b2	2	2022-05-18	(NULL)	1
a5	b1	1	2022-05-08	2	(NULL)
a5	b0	9	2022-05-18	1	(NULL)

Per la condizione del WHERE si eliminano i record con T.e=2 e il record con T.e=NULL, a causa dell'impossibilità di verificare la veridicità del predicato del WHERE.

Rimangono quindi 4 record come mostrato nel risultato seguente:

T				S	
a	b	c	d	e	f
a0	b1	9	2019-05-03	1	(NULL)
a1	b0	(NULL)	2019-05-01	1	6
a3	b0	9	2022-05-18	1	1
a5	b0	9	2022-05-18	1	(NULL)

# SOLUZIONI PRETEST

---

192) – b

La query considera i record che hanno T.c<9 e sono i seguenti:

a1	b1	0	2019-05-18	2
a3	b2	2	2022-05-18	(NULL)
a5	b1	1	2022-05-08	2

Attraverso una function che lavora su frame, i record vengono partizionati per il valore di T.b e ordinati sulla data (T.d).

Le partizioni ottenute riordinate, sono le seguenti:

a1	b1	0	2019-05-18	2
a5	b1	1	2022-05-08	2

a3	b2	2	2022-05-18	(NULL)
----	----	---	------------	--------

In queste partizioni si mettono i record che hanno data tra mese precedente e quello della current row.

Considero la partizione su b1 ordinata per d:

a1	b1	0	2019-05-18	2
a5	b1	1	2022-05-08	2

- Come current row prendo il primo record:

a1	b1	0	2019-05-18	2
----	----	---	------------	---

Tra la data di questo record (current row) e la data del suo mese precedente non ho altri record; quindi, il conteggio esterno mi darà 1

- Come current row prendo il secondo record:

a5	b1	1	2022-05-08	2
----	----	---	------------	---

Anche qui tra la data di questo record (current row) e la data del suo mese precedente non ho altri record, quindi il conteggio esterno mi darà 1

Considero ora la partizione su b2 ordinata per d (un record solo):

a3	b2	2	2022-05-18	(NULL)
----	----	---	------------	--------

Essendo che c'è un record solo, considerandolo come current row posso subito dire che, anche qui tra la data di questo record (current row) e la data del suo mese precedente non ho altri record; quindi, il conteggio esterno mi darà 1.

→ Tutti i valori di h sono  $\geq 1$ , e sono esattamente 1.

---

193) – a

La risposta b non vale perché, se prendo un insieme di dipendenze e applico la 3FN, guardando se a SX c'è una superchiave oppure se a DX ogni attributo è contenuto in una chiave → Ciò non ha nulla a che vedere con le caratteristiche della minimizzazione; quindi, il 'necessariamente' non è verificabile.

La risposta c no perché è il contrario della soluzione a.

# SOLUZIONI PRETEST

---

194) – b

La soluzione più efficiente è dettata dal gestore della base di dati, il quale riporta i risultati in memoria nel momento a lui più opportuno. Ha libero arbitrio. Nelle operazioni di recovery però, sorge il fatto dell'utilizzo degli insiemi UNDO e REDO delle operazioni, perché non sappiamo cosa effettivamente è stato portato in memoria.

---

195) – d

Si prende la tabella T e si fa una finestra fatta dalle partizioni di T.a che sono 2 (una per a1 e una per a2) ordinata su T.d in maniera decrescente, ottenendo il risultato seguente:

a	b	c	d
a1	b4	c3	4
a1	b2	(NULL)	2
a1	b1	c1	2
a2	b2	c2	5
a2	b3	(NULL)	1

La proiezione finale mi affianca ad ogni valore di T.a l'ultimo valore su T.d sulla partizione ottenendo:

a	d
a1	2
a2	1

---

196) – b

L'algebra relazionale è procedurale perché ci dice come un'interrogazione viene fatta cioè: in che ordine vengono fatte le operazioni e quali operatori si adoperano.

Il calcolo relazionale invece, ci dice solamente quale valore è atteso come risultato, ma non ci dice come si ottiene.

---

197) – b

La risposta c è falsa perché la chiave è formata da b1,b2 non ho chiavi esterne per poter mettere la cardinalità (1,1) → Quindi nemmeno la cardinalità è sbagliata.

La risposta b è giusta, perché non possiamo dire nulla sulla ridondanza senza sapere l'interpretazione dello schema E-R.

---

# SOLUZIONI PRETEST

---

198) – d

La soluzione è d per esclusione delle altre.

La risposta a è falsa perché:

- “Richiede il lock di ogni risorsa immediatamente prima di usarla” → Questo non dice nulla su entrambe le fasi del 2PL, cioè quando viene rilasciato il lock (prima/durante/dopo)
- “...non può richiedere il lock con troppo anticipo” → Non è detto il lock può essere preso anche all'inizio

La risposta b è falsa perché nel 2PL il deadlock viene generato.

La risposta c è falsa perché il fatto che nei 2PL i lock prima si prendono e poi si rilasciano. → Ciò non vuol dire che i lock subito dopo debbano essere ripresi, perché le fasi del 2PL sono due e non ce n'è una terza che rappresenta il riottenimento del lock.

---

199) – c

Prendiamo la tabella tt ridenominata in T, dove T.a<>a2 → Quindi rimangono i record dove T.a=a1.

a	b	c	d
a1	b2	(NULL)	2
a1	b4	c3	4
a1	b1	c1	3

Per ciascuno di questi record si proietta T.b e il valore dell'anticipo del record successivo di T.c ordinato su T.d (in modo crescente).

La risposta c è scelta per esclusione sulle altre.

La risposta a è falsa perché la query è corretta.

La risposta b è falsa perché l'ordinamento è suoi valori di T.d, il che non intacca il valore b2 della colonna T.b.

La risposta d è falsa perché in generale, le funzioni LEAD/LAG ad un certo punto avranno sempre un record con valore NULL, dato dal fatto che qualche volta non è presente il valore successivo/precedente!

La risposta e è falsa perché i record restituiti sono 3.

---

200) – c

È polinomiale perché si deve verificare soltanto se a SX delle dipendenze di R è presente una chiave/superchiave. → Il lavoro riguarda quindi, guardare tutte le parti di sinistra, a differenza della 3FN, dove si deve andare a destra anche tutti i possibili sottoinsiemi delle chiavi, che la rende con complessità esponenziale.

# SOLUZIONI PRETEST

---

201) – a

L'errore sta nel WHERE perché solitamente si scrive WHERE EXISTS ...., il T1.c non deve stare lì, in quel punto!

---

202) – b

La risposta d potrebbe essere giusta, ma la parola ‘solo’ implica che il trigger effettua quel tipo di operazione in maniera esclusiva, cosa che non è vera perché quel trigger potrebbe avere anche altri utilizzi → Si potrebbe non propagare l’aggiornamento, ma farci le computazioni sopra!

---

203) – b

Nel calcolo relazionale è possibile esprimere condizioni che non sono esclusivamente condizioni sui valori contenuti nella base di dati → Si possono scrivere cose ‘senza senso’. Nell’algebra relazionale invece, tutte le cose che si scrivono sono calcolate sul database, perché le espressioni sono indipendenti dal dominio degli attributi.  
→ Quindi, in questo caso tra il calcolo relazionale e l’algebra relazionale non è possibile un confronto.

---

204) – b

Le condizioni a SX e a DX dell’uguale sono equivalenti → Non c’è nessuna differenza rispetto all’ordine in cui si valutano le condizioni, visto che non butto via attributi, ma solo le righe.  
→ Fare prima cond o cond1 è irrilevante ai fini dell’uguaglianza, visto che i record che rimangono sono gli stessi.  
In fase di ottimizzazione, sarebbe più utile anticipare le selezioni ‘più selettive’, cioè quelle che buttano via più roba.

---

205) – b

Le variabili sono tutte diverse, indipendentemente da come si chiamino i domini.

---

206) – a

La risposta a è scelta per esclusione delle altre.  
Nella risposta b non ci sono tanti attributi quanti sono i valori assunti dai record di T sull’attributo t, ma ci sono tante colonne che hanno quel valore lì.

---

207) – a

# SOLUZIONI PRETEST

---

208) – b

Non le conserva le dipendenze, perché per le dipendenze  $\underline{CD} \rightarrow E$  e  $\underline{B} \rightarrow D$  non c'è una tabella dove possa verificare questi collegamenti.

---

209) – d

Le procedure non possono restituire dei result set ma solo variabili di uscita, che sono tutte scalari.

---

210) – a

Il concetto di transazione di per sé non genera anomalie, è la concorrenza tra le transazioni che le genera!

---

211) – b

Nella traduzione logica avrei:

- $A(a3, b1, b2, a1, a2, r) \rightarrow 1$  vincolo di integrità referenziale rappresentato dalla coppia  $b1, b2$  riferente alla tabella  $B$  e di conseguenza include anche l'associazione  $r1$ , perché non avrà una tabella separata
- $B(b1, b2, b3, b4) \rightarrow 0$  vincoli di integrità referenziale
- $C(c1, c2, c3) \rightarrow 0$  vincoli di integrità referenziale
- $S(a3, b1, b2, c1, s) \rightarrow 2$  vincoli di integrità referenziale rappresentati da  $a3, b1, b2$  per la tabella  $A$  e  $c1$  per la tabella  $C$

→ Totale 3 vincoli di integrità referenziale (1+0+0+2=3).

---

212) – d

Proietto il nome e cognome dei deputati, i cui codici sono diversi dai codici dei presidenti (di una commissione) → Si prendono i deputati che non sono presidenti (di commissione)  
→ Quindi, escludendo i presidenti, si considerano solo i deputati.

---

213) – a

---

214) – d

---

# SOLUZIONI PRETEST

---

215) – c

Risposta a falsa, perché nel WHERE non si mettono gli aggregati.

Risposta b falsa, perché per esempio si possono mettere aggregati anche nella SELECT.

Risposta d falsa, perché è assurdo mettere un aggregato nel SELECT per forza insieme ad un GROUP BY.

---

216) – c

---

217) – e

---

218) – c

Si esegue una differenza, perché si considera tutta la tabella M, e di questa tabella non si prende la parte di un'altra tabella M2 (identica), che lo stesso valore dell'attributo a, ma ha valore diverso sull'attributo e.

OSS: Un possibile doppio NOT EXISTS ci potrebbe portare ad avere una divisione insiemistica.

---

219) – a

---

220) – c

Dato che ho l'AND considero le selezioni separatamente.

Inizio verificando la selezione, su quella più selettiva (in questo caso è indifferente, inizio con A).

A='n1' su R data da → 50000/50=1000 elementi.

Su questi 1000 elementi ci vado a vedere la parte dell'AND cioè B='n2'; quindi, se ho elementi A='n1' con B='n2'.

È possibile che i B siano tutti uguali a 'n2' quando ho A='n1' con 1000 elementi?

Sì, al massimo sono 1000 elementi anch'essi.

Potrei buttare via qualcosa ma mi serve il caso peggiore, cioè 1000 elementi.

Facendo la proiezione di A non ho nulla di nuovo ma con quella di C sì.

→ Considero che i C sono 1000 elementi e possono essere tutti diversi da A ma, per come A è stato ottenuto dalla selezione, decido di non buttare via nulla e, mi tengo una tabella di C come su A, cioè C di 1000 elementi.

Per la proiezione di A e C ho che entrambi occupano una parola e quindi in totale sono 1000\*2=2000 elementi.

---

# SOLUZIONI PRETEST

---

221) – c

L'attributo riferito cioè a (in questo caso), farà parte della chiave nell'entità riferente.

---

222) – c

---

223) – a

Per cammino di esecuzione di una transazione si intende il percorso dal suo inizio (start-transaction) alla sua fine (end-transaction) e, all'interno di questo cammino deve essere eseguito una sola volta uno dei due seguenti comandi:

- Commit work
  - Rollback work (abort)
- 

224) – c

Considero nome e cognome dei deputati:

$\Pi_{\text{nome}, \text{cognome}}(\text{Deputato})$

- (meno)

Un result set vuoto perché la seguente query non proietta nulla, visto che il JOIN non produce niente di sensato → Combinazione tra una selezione sulle Commissioni Esteri (delle commissioni) con una selezione sulle Commissioni Nulle (dei deputati).

```

$$\begin{array}{l} \Pi_{\text{nome}, \text{cognome}}( \\ \quad \Pi_{\text{numero}}(\sigma_{\text{nome}='esteri'}(\text{Commissione})) \\ \quad \bowtie_{\text{numero}=\text{commissione}} \\ \quad \Pi_{\text{commissione}, \text{nome}, \text{cognome}}(\sigma_{\text{commissione=null}}(\text{Deputato})) \\ \end{array}$$

```

---

225) – d

La soluzione è la risposta d anche se è ambigua, dato che non viene rappresentato il fatto che le province, sono pure quelle della Lombardia oltre a non essere quelle della Toscana. In questo caso prendo il nome delle province non in Toscana, perché l'OR mi impone di scegliere o l'una o l'altra condizione, e nel testo la risposta era sulla seconda condizione. Un altro esercizio dello stesso esame, con il testo identico a questo ma con risposte diverse che riguardavano la prima condizione dell'OR, aveva la soluzione:  
“Elenca le province della Lombardia in cui sono stati eletti deputati”.

---

# SOLUZIONI PRETEST

---

226) – c

*Soluzione ottenuta per esclusione delle altre.*

*La risposta a no, perché lo schedule che usa il protocollo TS è view-equivalente allo schedule che riproduce l'ordine delle richieste delle transazioni (schedule seriale), dato che quest'ultimo è accettato solo se riflette l'ordine seriale delle transazioni indotte dal protocollo TS.*

*Soluzione b no, perché i protocolli 2PL e TS sono incomparabili sulla view-equivalenza.*

*Soluzione d no, perché gli schedule che usano il protocollo TS sono automaticamente CSR, di conseguenza vale la view-equivalenza da TS verso CSR.*

---

227) – c

*Soluzione ottenuta per esclusione delle altre.*

*La risposta a no perché uno schedule 2PL è equivalente ad uno schedule CSR.*

*La risposta b no, perché lo schedule che usa il protocollo 2PL è view-equivalente allo schedule che riproduce l'ordine delle richieste delle transazioni (schedule seriale).*

*Soluzione d no, perché i protocolli 2PL e TS sono incomparabili, sulla view-equivalenza.*

---

228) – d

*Soluzione ottenuta per esclusione delle altre per il seguente motivo*

- Quando una transazione abortisce (abort) tutte le sue operazioni vengono disfatte (UNDO) e il record di abort viene scritto nel log, senza togliere le operazioni originali, perché il log deve tenere traccia di tutti i passaggi delle transazioni
- 

229) – b

*Nella soluzione “...al più una...” → Impone al massimo una, riferito alla cardinalità massima.*

---

230) – b

*Soluzione ottenuta per esclusione delle altre.*

*La risposta a è falsa, perché per esempio ho il DUMP che viene utilizzato solo nella ripresa a freddo e non nella ripresa a caldo.*

*La risposta c è falsa, perché il BEFORE STATE si trova prima dello svolgimento delle relazioni, e può non avere in comune niente con quello che c'è nel DUMP.*

*La risposta d è falsa, perché si parte dal BEGIN delle transazioni che vanno disfatte o rifatte.*

---

231) – c

---

# SOLUZIONI PRETEST

---

232) – a

'...la potenziale...' perché il record da prelevare potrebbe esserci oppure no.  
Se c'è la variabile classica FINITO rimane 0, e quindi si procede con il flusso di esecuzione.  
Se la variabile non c'è si mette la SIGNAL.

---

233) – d

Non compila perché, innanzitutto per il cursore mancano la OPEN, la CLOSE e la FETCH e inoltre, è presente una INTO dentro la SELECT.

---

234) – d

Fasi dell'espressione:

- Si prendono le province e i collegi dei deputati tali che  
provincia: sg, collegio: cg
  - Si considerano tutti i deputati dello stesso collegio di provincia, con lo stesso nome e cognome → JOIN su nome:n= nome:n, cognome:c=cognome:c,  
provincia:sg=provincia:sg, collegio:cg=collegio:cg delle due tabelle Deputato  

Deputato(codice:cd, nome:n, cognome:c, commissione:num, provincia: sg, collegio: cg)  $\wedge$  Deputato(codice:cd', nome:n, cognome:c, commissione:num', provincia: sg, collegio: cg)
  - Che hanno codice diverso, cioè sono deputati diversi (per non far riferimento allo stesso deputato)  
cd ≠ cd'
- 

235) – c

La selezione su A='n1' su R1 è data da → 50000/500=100 elementi.

La selezione su B='n2' su R2 è data da → 50000/500=100 elementi.

Il NATURAL JOIN è fatto sull'attributo B in comune e per fare il JOIN devo valutare la tabella R1.

Per la tabella R1 ci sono due casi:

1. Non ho elementi di R1 con B='n2'
2. Ho un solo elemento di R1 con B='n2'

Quindi, avrò al massimo 1 elemento di R2 con B='n2' e A='n1' (dalla selezione), questo perché le tabelle per definizione non hanno righe uguali.

Questo elemento sarà l'unico che farà JOIN con gli elementi selezionati su R2, che sono 100  
→ Avrò quindi una tabella con gli attributi A, B, C con 100 elementi.

# SOLUZIONI PRETEST

---

236) – b

Ragionamento simile alla soluzione dell'esercizio [245](#) solo che, qua per i 'periodi interamente contenuti nel periodo di una tariffa' considero, oltre che la proiezione su DataInizioPrenotazione anche la proiezione su DataFinePrenotazione:

$\pi_{\text{DataInizioPrenotazione}, \text{DataFinePrenotazione}, \text{IDPrenotazione}}(\sigma_{\text{CodFiscaleCliente} = 'XXX'} (\text{Prenotazioni}))$

Che poi va a combinarsi con DataInizioTariffa e DataFineTariffa così:

☒ DataInizioPrenotazione  $\geq$  DataInizioTariffa and DataFinePrenotazione  $\leq$  DataFineTariffa  
Tariffa

---

237) – b

Un insieme minimale si ha quando:

- Non si possono togliere attributi a sinistra della freccia
- Non si possono togliere dipendenze

Ciò esclude le risposte a, c, d.

---

238) – b

Spiegazione della query:

- La WHERE mi considera il mese >5 della colonna T.d e mi rimangono solamente questi due record

T				
a	b	c	d	e
a3	b2	2	2022-06-18	2
a5	b0	3	2022-07-18	1

- Proietto poi su T.c (che sono il 2 e il 3) a ciascuno dei quali viene affiancato il RANK, cioè la sua posizione, in una classifica stilata partizionando i record per T.e e ordinando per T.c → Le partizioni per questi due record sono due partizioni diverse (2!=1), quindi entrambi i record avranno 1 come posizione in classifica, della propria partizione → Viene fatto il 'restart numbering' del RANK, visto che questo ha valore nella partizione

c
2
3

# SOLUZIONI PRETEST

---

239) – b

La risposta a è falsa perché se tolgo ORDER BY T.c viene considerata per la somma tutta la tabella.

La risposta c è falsa, perché per fare ciò che dice la risposta, ci sarebbe dovuto essere PARTITION BY T.a.

La risposta d è falsa, perché la somma viene fatta e quindi non è necessaria PARTITION BY.

---

240) – b

Se considero il deferred refresh di tipo FULL può essere fatto senza la log table, from scratch, quindi non è detto che si utilizzi una log table.

---

241) – d

Per definizione.

---

242) – b

Fasi dell'espressione:

- Si prendono i nomi delle commissioni tali che

nome:nc

- Si considera tra i deputati quello che si chiama Mario Rossi

Deputato(codice:cd, nome:n, cognome:c, commissione:num, provincia: sg, collegio: cg)  $\wedge$  n='Mario'  
 $\wedge$  c='Rossi'

- Il quale deputato è presidente di Commissione

Deputato(codice:cd, nome:n, cognome:c, commissione:num, provincia: sg, collegio: cg)  $\wedge$  n='Mario'  
 $\wedge$  c='Rossi'  $\wedge$  Commissione(numero:nm, nome:nc, presidente:cd)}

JOIN tra codice:cd della tabella Deputato e presidente:cd della tabella Commissione

OSS: La risposta dice "...il nome della commissione..." anche se potrebbe essere pure al plurale "...i nomi delle commissioni..."

---

243) – a

I database che scalano in modo orizzontale, con l'aggiunta di più server sono quelli NoSQL. Questi database sono strutturati secondo una rete di server (standard), e di conseguenza se ne possono aggiungere altri, di server standard.

Questa operazione è dovuta al fatto di avere tanti server più o meno equivalenti.

La risposta c no, perché si possono anche usare dei server più potenti, ma in generale è necessario aumentare il numero dei server, non la potenza.

---

# SOLUZIONI PRETEST

---

## 244) – b

La selezione su A='n1' su R1 è data da → 50000/500=100 elementi.

La selezione su B='n2' su R2 è data da → 50000/500=100 elementi.

Il NATURAL JOIN è fatto sull'attributo B in comune.

Per definizione sulle tabelle che non hanno righe uguali, avrò al massimo 1 elemento di R1 con B='n2' e A='n1'.

Questo elemento si combinerà tramite il NATURAL JOIN, con i valori di B presenti in R2, che saranno al massimo 100 anch'essi.

Questa affermazione sui 100 valori di B di R2 è dovuta al fatto che, dai dati iniziali, i valori diversi di B sono 500 e la tabella ha un numero di tuple 5000 (quindi è sottointeso il calcolo 50000/500=100 elementi).

---

## 245) – a

SPIEGAZIONE espressione algebrica:

- Si prendono le prenotazioni fatte da una persona il cui codice fiscale è 'XXX'

$\sigma_{\text{CodFiscale}='XXX'}$  (Prenotazioni)

si combinano con le prenotazioni delle stanze tramite il seguente JOIN



PrenotazioneStanza

e se ne prendono la data di inizio prenotazione e l'id di prenotazione  
(mi serve per il JOIN)

$\pi_{\text{DataInizioPrenotazione}, \text{IdPrenotazione}}$

- Di questo risultato precedente si prende la data di inizio prenotazione  
(mi serve per il JOIN) e si mette in combinazione con il seguente risultato
- Si confronta la data di inizio della prenotazione con la data di inizio della tariffa e la data della fine tariffa

$\bowtie_{\text{DataInizioPrenotazione} \geq \text{DataInizioTariffa} \text{ and } \text{DataInizioPrenotazione} \leq \text{DataFineTariffa}}$   
Tariffa

- Alla fine, l'espressione algebrica produce il numero della stanza dei risultati precedentemente ottenuti

$\pi_{\text{NumStanza}}$

→ Si ottengono le stanze prenotate da 'XXX', relative ad una data d'inizio di prenotazione, che rientrano completamente in una data di tariffa (che è la risposta a).

# SOLUZIONI PRETEST

---

246) – d

*Soluzione scelta per esclusione delle altre risposte.*

*La risposta b è sbagliata perché, se considero in una proiezione sugli attributi di raggruppamento un altro attributo, per esempio un aggregato → Non è detto che abbia per forza dei duplicati.*

*La risposta c è sbagliata perché, se per esempio considero le combinazioni tra medici e pazienti, raggruppando per cognome (che non è chiave) dei pazienti senza il DISTINCT e proiettandoli con la somma delle parcellle dei medici → Potrei avere due pazienti con lo stesso cognome che hanno speso entrambi la stessa cifra, perché ho raggruppato su un attributo non chiave, quindi ho un duplicato.*

*La risposta a è assurda.*

---

247) – a

*La WHERE è verificata quando il valore di T.c è minore di tutti i valori provenienti dalla subquery.*

*La subquery proietta tutti i valori di T.c laddove sono diversi dal valore T.c della query esterna.*

*Quindi si considerano i valori T.c della query esterna minori di tutti i valori T.c della subquery → Quindi si considerano i valori di T.c minori di tutti i valori T.c tranne sé stesso.  
Alla fine, si proietta il valore di T.d.*

*Per la WHERE (query esterna) innanzitutto si prende il valore che ha T.c=0, che è il più piccolo di tutti e inoltre, viene preso anche il valore di T.c=NULL.*

*Si considera pure il record con T.c=NULL perché il NULL<ALL(subquery) è vero a prescindere. Questa affermazione è vera perché non ho un termine di paragone dalla subquery, poiché la subquery mi restituisce un result set vuoto e non potendo confrontare ho la condizione vera in partenza → Il result set vuoto è dovuto alla condizione della subquery TT.c>T.c, che è sempre falso, dato che il T.c (query esterna) è NULL, e di conseguenza il NULL non è confrontabile → Quindi result set vuoto.*

*Alla fine, proiettando T.d avrò il seguente risultato:*

T
d
2019-05-01
2019-04-18

# SOLUZIONI PRETEST

---

248) – a

Le soluzioni c e d sono sbagliate, perché per esempio se prendiamo in considerazione la FORCE, questa va eseguita indipendentemente dal carico applicativo.

Questa istruzione deve essere eseguita da parte del gestore, nel momento di trasferimento dei dati riguardanti le transazioni, in memoria secondaria.

---

249) – a

Per avere l'insieme minimale dobbiamo vedere se si possono togliere attributi a SX dalle varie dipendenze.

Qua ho due dipendenze che hanno due attributi a SX: AB → C e AB → E.

Bisogna vedere se è possibile arrivare a C o ad E partendo da A o da B, senza usare l'altro attributo.

Se faccio la chiusura di B, vedo che contiene CDE → Quindi da B si può andare a C e ad E senza A, di conseguenza A si toglie e ci rimangono le seguenti dipendenze:

- B → C B → D D → C C → E B → E

Qua vediamo inoltre, che ci sono delle dipendenze inutili come B → C e B → E perché partendo da B → D posso arrivare sia a C che ad E, con un passaggio in più, quindi con il cammino più lungo (quello che ci interessa) → Allora si tolgonon queste due dipendenze inutili e si ottiene la soluzione a.

---

250) – c

Essendo la cardinalità massima (N,N) tra le due entità, viene definita una tabella per R.

Questa tabella conterrà le chiavi delle due entità + i suoi attributi → In R avrò sempre la coppia Ka,Kb che rappresenterà gli elementi esistenti, fra le due entità.

Inoltre, avrò anche due tabelle per le due entità A e B.

# EXTRA (PROCEDURE PER ESERCIZI)

---

## EXTRA 1: NON APPARTENENZA IN ALGEBRA!

Elencare tutte le tuple che non hanno X

→ Equivale a dire tutte le tuple – tutte le tuple che hanno X

---

## EXTRA 2: DIVERSITÀ NEL JOIN IN ALGEBRA!

```
πnome,cognome(πnumero(σnome="Bilancio"(Commissione))  
▷commissione ≠ numero  
πcommissione,nome, cognome(Deputato))
```

Elencare i deputati membri di una commissione diversa dalla Commissione Bilancio

---

## EXTRA 3: DIVERSITÀ NEL JOIN SUI DOMINI!!

```
{nome:n, cognome:c | Commissione(numero: num, nome: nc, presidente:pr) ∧ Deputato(codice:cd, nome:n, cognome:c,  
commissione:num', provincia:pv, collegio:cg) ∧ nc='Esteri' ∧ num ≠ num'}
```

Elencare i deputati presenti in commissioni diverse dalla Commissione Esteri

---

## EXTRA 4: DIVERSITÀ NEGLI ATTRIBUTI SUI DOMINI!!

```
{nome:np | Provincia(sigla: sg, nome: np, regione: cr) ∧ Deputato(codice:cd, nome:n, cognome:c, commissione:num,  
provincia: sg, collegio: cg) ∧ Regione(codice: cr, nome:nr) ∧ (nr ≠ 'Sicilia' ∧ nr ≠ 'Toscana'))}
```

Elencare i nomi delle province non della Sicilia né della Toscana in cui c'è almeno un deputato eletto

---

## EXTRA 5: NON ESISTENZA SUI DOMINI!!

```
1. πAB(R) – πAB(R ▷◁C=D S)  
2. {A:a, B:b | not exists e,c . R(A:a, B:b, C:c) and S(D:c, E:e) }
```

```
1. πAB(R) – πAB(R ▷◁C=D S)  
2. {A:a, B:b | R(A:a, B:b, C:c) and not exists e . S(D:c, E:e) }
```

AND e NOT EXISTS Sono interscambiabili, però attenzione:

- Nel 2° non esiste la variabile della relazione S quindi il JOIN va a vuoto senza produrre nulla
  - Nel 1° il JOIN va a VUOTO perché non esistono entrambe le variabili della relazione S
-

# EXTRA (PROCEDURE PER ESERCIZI)

---

## EXTRA 6: RIDENOMINAZIONI!

1° CASO

$$(R_3 \triangleright \triangleleft_{G=E} R_2) \cup (\rho_{DG \leftarrow AC}(\pi_{ACEF}(R_1 \triangleright \triangleleft_{B=F} R_3)))$$

$$\{ D: d, G: g, E: e, F: f \mid R_3(E: e, F: f) \wedge ((R_2(D: d, G: g) \wedge (g=e)) \vee (R_1(A: d, B: b, C: g) \wedge (b=f))) \}$$

2° CASO

$$\rho_{E,F \leftarrow A,B}(\pi_{A,B,D}(R_1 \triangleright \triangleleft R_2))$$

$$\{D:d, E:a, F:b \mid R_1(A:a, B:b, C:c, D:d) \text{ and } R_2(B:b, D:d) \}$$

→ Nel 1° caso la ridenominazione  $DG \leftarrow AC$  la faccio nella relazione R1 interna dopo il '|', perché, il JOIN è fatto su B e F e quindi non ho bisogno di A e C e posso cambiarli subito

→ Nel 2° caso invece la ridenominazione  $EF \leftarrow AB$  è fatta prima di '|', perché il JOIN è naturale ed è fatto sugli attributi comuni ad R1 e R2, e R1 ed R2 hanno in comune **d**, che non ha bisogno di ridenominazione **e** e **b**, che va ridenominato; se lo ridenomino dopo il '| il NATURAL JOIN non viene fatto e quindi sarebbe un errore

---

## DIPENDENZE FUNZIONALI REGOLE EQUIVALENZA:

Se data una dipendenza **F** si chiede di trovare la dipendenza equivalente **G**, e quest'ultima ha le parti sinistra ridotte, senza attributi estranei

→ Conviene utilizzare la chiusura transitiva sull'insieme degli attributi  $(X_i)^{+G}$  e  $(X_j)^{+F}$   
 $X_i$  appartengono a **F** e  $X_j$  appartengono a **G**

Altrimenti, nel caso fossero ridotte solo le parti sinistre

→ Conviene usare l'algoritmo di minimizzazione

# EXTRA (PROCEDURE PER ESERCIZI)

---

## DIPENDENZE FUNZIONALI CHIUSURA SULL'INSIEME DI ATTRIBUTI:

Per fare la chiusura transitiva sugli insiemi di attributi devo fare  $(X_i)^{+G} = (X_j)^{+F}$  con gli  $X_i$  che appartengono a F e gli  $X_j$  appartengono a G

- Considero il seguente insieme di dipendenze F e verifico quale dei successivi insiemi G è equivalente ad F

F :  $AB \rightarrow C, AD \rightarrow EF, B \rightarrow A, A \rightarrow D, B \rightarrow E$

G1 :  $B \rightarrow A, A \rightarrow DEF$

G2 :  $B \rightarrow C, A \rightarrow DF, B \rightarrow A, B \rightarrow E$

G3 :  $B \rightarrow AC, A \rightarrow DEF, B \rightarrow E$

---

- Inizio facendo la chiusura di  $(X)^{+G_1}$  con gli attributi X che riguardano quelli dell'insieme di dipendenze di F
  - $AB \rightarrow C, (AB)^{+G_1} = ABDEF$ , C non compare in questa sequenza, quindi non appartiene ad  $(AB)^{+G_1}$ , mi fermo qua perché la chiusura di  $(X)^{+G_1}$  non mi produce l'equivalenza
- Inizio facendo la chiusura di  $(X)^{+G_2}$  con gli attributi X che riguardano quelli dell'insieme di dipendenze di F
  - $AB \rightarrow C, (AB)^{+G_2} = ABCDEF$ , C compare in questa sequenza, quindi appartiene ad  $(AB)^{+G_2}$ , vado avanti con i successivi attributi
  - $AD \rightarrow EF, (AD)^{+G_2} = ADF$ , F compare in questa sequenza ma E no, quindi EF non appartiene ad  $(AD)^{+G_2}$ , mi fermo qua perché la chiusura di  $(X)^{+G_2}$  non mi produce l'equivalenza
- Inizio facendo la chiusura di  $(X)^{+G_3}$  con gli attributi X che riguardano quelli dell'insieme di dipendenze di F
  - $AB \rightarrow C, (AB)^{+G_3} = ABCDEF$ , C compare in questa sequenza, quindi appartiene a  $(AB)^{+G_3}$ , vado avanti con i successivi attributi
  - $AD \rightarrow EF, (AD)^{+G_3} = ADEF$ , EF compare in questa sequenza, quindi appartiene a  $(AD)^{+G_3}$ , vado avanti con i successivi attributi
  - $B \rightarrow A, (B)^{+G_3} = ABCDEF$ , A compare in questa sequenza, quindi appartiene a  $(B)^{+G_3}$ , vado avanti con i successivi attributi
  - $A \rightarrow D, (A)^{+G_3} = ADEF$ , D compare in questa sequenza, quindi appartiene a  $(B)^{+G_3}$ , vado avanti con i successivi attributi
  - $B \rightarrow E, (B)^{+G_3} = ABCDEF$ , E compare in questa sequenza, quindi appartiene a  $(B)^{+G_3}$ , mi fermo perché ho finito gli attributi
- Ho trovato quindi che la chiusura di  $(X)^{+G_3}$  con gli attributi X che riguardano quelli dell'insieme di dipendenze di F, mi può produrre equivalenza, ora devo fare il viceversa

# EXTRA (PROCEDURE PER ESERCIZI)

---

## ... CONTINUO DIPENDENZE FUNZIONALI CHIUSURA SULL'INSIEME DI ATTRIBUTI:

- Adesso faccio la chiusura di  $(X)^+F$  con gli attributi X che riguardano quelli dell'insieme di dipendenze di G3
  - $B \rightarrow AC$ ,  $(B)^+F = ABCDEF$ , AC compare in questa sequenza, quindi appartiene a  $(B)^+F$ , vado avanti con i successivi attributi
  - $A \rightarrow DEF$ ,  $(A)^+F = ADEF$ , DEF compare in questa sequenza, quindi appartiene a  $(A)^+F$ , vado avanti con i successivi attributi
  - $B \rightarrow E$ ,  $(B)^+F = ABCDEF$ , E compare in questa sequenza, quindi appartiene a  $(B)^+F$ , mi fermo perché ho finito gli attributi
- Ho trovato quindi che la chiusura di  $(X)^+F$  con gli attributi X che riguardano quelli dell'insieme di dipendenze di G, mi può produrre equivalenza
- Considerando  $(X)^{+G3}$  come  $(Xj)^{+G3}$  e  $(X)^+F$  come  $(Xi)^+F$  ottengo che  $(Xi)^{+G3} = (Xj)^+F$  e di conseguenza ho verificato l'equivalenza

OSS: Sarebbe consigliato di minimizzare le due dipendenze funzionali prima di farci la chiusura sull'insieme di attributi

---

## ALGORITMO DI MINIMIZZAZIONE

- 1) Porto tutte le dipendenze con un solo valore alla parte DESTRA, utilizzando le regole di ARMSTRONG che spezzano le dipendenze
- 2) Porto le dipendenze in forma canonica, togliendo a SX gli attributi inutili per arrivare all'attributo successivo oppure togliendo agli attributi di SX gli attributi dai quali dipendono  
ES. Data la dipendenza **AB**→**X** si possono verificare due casi, scorrendo i successivi attributi:
  - a. A→B, cioè B dipende da A e si toglie dalla dipendenza iniziale la lettera B, quindi ottengo **A**→**X**, **A**→**B**, ...
  - b. B→A, cioè A dipende da B e si toglie dalla dipendenza iniziale la lettera A, quindi ottengo **B**→**X**, **B**→**A**, ...
- 3) Controllo se ci sono dipendenze RIDONDANTI, cioè più di una dipendenza che mi fa arrivare da un punto (iniziale) ad un altro (finale), e scelgo di togliere la dipendenza che mi fa fare la strada più corta
- 4) Se necessario riunisco alcune dipendenze, dipendendo da come sono poste le risposte dei testi degli esercizi perché potrei avere
  - a. Caso in cui le dipendenze dopo esser state spezzate si riuniscono
  - b. Caso in cui le dipendenze rimangano divise

# EXTRA (PROCEDURE PER ESERCIZI)

## APPLICAZIONE ALGORITMO DI MINIMIZZAZIONE

- 
- Dato l'insieme di dipendenze funzionali:  $AB \rightarrow C, B \rightarrow A, C \rightarrow E, AB \rightarrow E$ 
    - 1)  $AB \rightarrow C, B \rightarrow A, C \rightarrow E, AB \rightarrow E$ , rimane così com'è perché a DX ho solo un elemento
    - 2) Quà ho che A dipende da B, quindi, tolgo A dalla dipendenza nella parte SX ed ottengo:  $B \rightarrow C, B \rightarrow A, C \rightarrow E, B \rightarrow E$
    - 3) Mi accorgo che B è l'attributo di partenza per arrivare alla fine ad E, e la strada più corta per arrivare ad E è tramite la dipendenza  $B \rightarrow E$  che è una ridondanza, quindi la tolgo ed ottengo  $B \rightarrow C, B \rightarrow A, C \rightarrow E$
    - 4) Poiché l'esercizio ha le soluzioni con le dipendenze divise lo lascio così com'è e quindi la dipendenza equivalente minima ottenuta è  $B \rightarrow C, B \rightarrow A, C \rightarrow E$
  
  - Dato l'insieme di dipendenze funzionali:  $A \rightarrow BCD, CD \rightarrow E, B \rightarrow D, A \rightarrow E$ 
    - 1)  $A \rightarrow B, A \rightarrow C, A \rightarrow D, CD \rightarrow E, B \rightarrow D, A \rightarrow E$
    - 2) Nell'unica dipendenza che ha 2 attributi alla sua sinistra non ho dipendenze che dipendano da altre, quindi lascio così com'è al punto 1)
    - 3) Mi accorgo che l'attributo di partenza è A e quello di arrivo è E, e per arrivare da A ad E la strada più corta è  $A \rightarrow E$ , che è una ridondanza quindi la tolgo. Mi accorgo inoltre, che partendo da A per arrivare a D (anche se non è il punto finale), ho due percorsi e quello più corto è  $A \rightarrow D$ , e quindi un'altra ridondanza, la tolgo. Ottengo quindi l'insieme di dipendenze così:  $A \rightarrow B, A \rightarrow C, CD \rightarrow E, B \rightarrow D$
    - 4) Poiché l'esercizio ha le soluzioni con le dipendenze unite, unisco  $A \rightarrow B$ , e  $A \rightarrow C$  ed ottengo la dipendenza equivalente minima:  $A \rightarrow BC, CD \rightarrow E, B \rightarrow D$

---

## GRAFICO TECNICHE DI GESTIONE CONCORRENZA

