Nome e Cognome:

Data:

* **Non so**

1. Quale simbolo si utilizza per concatenare le stringhe? ……………….

* **Non so**

1. A cosa serve l'operatore **!** ? ……………….……………….

* **Non so**

1. Una classe non può essere dichiarata:
2. Static
3. Private
4. Default
5. Nessuna delle precedenti

* **Non so**

1. Come si può evitare che una variabile membro diventi serializzata?
2. Marcandola private
3. Marcandola volatile
4. Marcandola transient
5. Non è possibile
6. Cos'è un instanceof?

* **Non so**

1. I metodi di un oggetto
2. Un operatore e parola chiave
3. Nessuna delle precedenti
4. Il seguente frammento di codice java:

**class A {**

**public static void main(String [] args) {**

**B b = new A();**

**}**

**}**

**class B extends A {}**

* **Non so**

Quale risultato dà?

1. Compile error
2. Runtime Exception
3. No error
4. Il seguente frammento di codice java:

**int a1 = 5; double a2 = (float)a1;**

* **Non so**

Quale risultato dà?

1. Compilation error
2. Runtime error
3. No errors
4. Il seguente frammento di codice java:

**float a = 9/0;**

* **Non so**

Quale risultato dà?

1. Compilation error: Divisions must be in a try block
2. Compilation error: DivideByZeroException
3. Runtime Exception
4. No Error: a is NaN
5. Il seguente frammento di codice java:

**int a = 9/0;**

* **Non so**

Quale risultato dà?

1. Compilation error: Divisions must be in a try block
2. Compilation error: DivideByZeroException
3. Runtime Exception
4. No Error: a is NaN
5. Il seguente frammento di codice java:

**class A {**

**int b = 1;**

**public static void main(String [] args) {**

**System.out.println("b is " + b);**

**}**

**}**

* **Non so**

Quale risultato dà?

1. Compilation error
2. Runtime Error
3. Runtime Exception
4. Output of b is 1
5. Il seguente frammento di codice java:

**class A {**

**public static void main(String [] args) {**

**A a = new B();**

**}**

**}**

**class B extends A {}**

* **Non so**

Quale risultato dà?

1. Compiler error
2. Runtime Exception
3. No errors
4. Il seguente frammento di codice java:

**int a = 3.5;**

* **Non so**

Quale risultato dà?

1. Compilation error
2. Runtime error
3. a being 3.5
4. a being 3
5. Il seguente frammento di codice java:

**String s = "21.0";**

**Double y;**

**if (s != null)**

**{**

**double x = Double.parseDouble(s);**

**y = x / 3.0;**

**}**

**System.out.println("risultato: " + y);**

* **Non so**

Quale risultato dà?

1. compila ed esegue correttamente, stampando la stringa "risultato: 7.0"
2. errore in compilazione
3. compila correttamente ma l'esecuzione provoca la generazione di un’eccezione
4. nessuna delle precedenti
5. Il seguente frammento di codice java:

**Int k = 4;**

**int n = k;**

**int p = 1;**

**while (n > 0)**

**{**

**p \*= n;**

**n--;**

**}**

**System.out.println(k + "! = " + p);**

* **Non so**

Quale risultato dà?

1. Il frammento di codice compila, ma in esecuzione ciclo all'infinito
2. Il frammento di codice compila e produce a standard output la stringa "4! = 24"
3. Il frammento di codice non compila, il ciclo while deve essere sostituito da un ciclo for
4. Nessuna delle precedenti affermazioni è corretta
5. Il seguente frammento di codice java:

**char c1 = 'A';**

**c1++;**

**System.out.println(c1);**

Quale risultato dà?

* **Non so**

1. il codice stampa su standard output il carattere 'B'
2. errore di conversione in compilazione: char c1 = 'A' + 1; 'A' + 1 non e' un carattere
3. errore in esecuzione: java.lang.NumberFormatException: 'A' + 1 non e' un carattere
4. nessuna delle precedenti
5. Quale è il risultato dell'esecuzione del seguente metodo java?

**public static int fattoriale(int n) {**

**int p = 1;**

**while (n > 0) {**

**p = p \* n;**

**n = decrementa(n);**

**}**

**return p;**

**}**

* **Non so**

OPZIONI:

1. il fattoriale di n se n > 0, altrimenti 1
2. sempre 1, per qualsiasi valore di n
3. il metodo non termina a causa di un ciclo infinito
4. Il seguente frammento di codice java:

**int a[] = {1, 1, 2, 3, 5, 8, 13, 21};**

**int i = a.length - 1;**

**int target = 8;**

**while (i >= 0) {**

**if (a[i] == target) break;**

**i--;**

**}**

**System. out.println("a[" + i +"] = " + target);**

* **Non so**

Quale risultato dà?

1. l'esecuzione del frammento non termina
2. stampa la stringa "a[5] = 8"
3. errore di compilazione
4. stampa la stringa "a[4] = 8"
5. nessuna delle precedenti risposte è corretta
6. Dato il seguente frammento di codice java:

**public class Test {**

**public static void main(String[] args {**

**Integer x= new Integer (5) ;**

**Integer y= new Integer (5) ;**

**System.out.println (x==y) ;**

**}**

**}**

* **Non so**

Che cosa restituisce x==y?

1. Errore di compilazione
2. Runtime Exception
3. True
4. False
5. Quali, tra i seguenti frammenti di codice javascript, attribuiscono alla variabile b un valore maggiore di 0, assumendo che i valori di a e c siano rispettivamente 5 e 0?
6. if(a > 0) if(c > 0) b=100;
7. if((c >= a) || (c >= 0)) b=100; else b=-100;
8. if(a>c && c>=0) b=-100; else b=100;
9. if(c>0) if (a>4) b=100 else b=-100;

* **Non so**

1. Qual è, tra i seguenti, il modo corretto per riferirsi ad un foglio di stile chiamato “style.xls”?
2. <stylesheet type="text/xsl" href="style.xsl" />
3. <link type="text/xsl" href="style.xsl" />
4. <?xml-stylesheet type="text/xsl" href="style.xsl" ?>

* **Non so**

1. Dato il seguente frammento di foglio di stile:

* **Non so**

**tr.a { background-color: green;}**

**tr.b {color : red;}**

**span.c {color : blue;}**

Modificare il seguente frammento di codice HTML in modo che:

- la prima riga della tabella abbia sfondo verde

- il testo nella seconda riga della tabella sia rosso, ad eccezione della prima cella che deve avere testo di

colore blue

|  |  |
| --- | --- |
| Codice originale | Codice modificato |
| <table>  <tr>  <td> Laura </td>  <td> Verdi </td>  </tr>  <tr>  <td> Azzurra </td>  <td> Rossi </td>  </tr>  <tr>  <td> Francesca </td>  <td> Incolore </td>  </tr>  </table> |  |

1. Dato il seguente schema relazionale:

* **Non so**

**STUDENTI (Matricola, NomeS, CorsoLaurea\*, AnnoN)**

**CORSIDILAUREA (CorsoLaurea, TipoLaurea, Facoltà)**

**FREQUENTA (Matricola\*, CodCorso\*)**

**CORSI (CodCorso, NomeCorso, CodDocente\*)**

**DOCENTI (CodDocente, NomeD, Dipartimento)**

Scrivere le interrogazioni SQL che restituiscono le seguenti informazioni:

1- Il nome e l’anno di nascita degli studenti iscritti al corso di laurea SBC, in ordine rispetto al nome

2- Il nome ed il Dipartimento dei docenti di “Basi di Dati e Sistemi Informativi” o di

“Informatica Generale”

3- Matricola e nome degli studenti di un corso di laurea triennale (tipoLaurea = 'L') che

seguono un corso di un docente di nome Felice.

4- Per ogni tipo di laurea, il tipoLaurea e l’età media degli studenti

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Database**

1. Row is synonymous with the term:
   1. Record.
   2. relation.
   3. column.
   4. field.
2. Which of the following is a group of one or more attributes that uniquely identifies a row?
   1. Key
   2. Determinant
   3. Tuple
   4. Relation
3. When the values in one or more attributes being used as a foreign key must exist in another set of one or more attributes in another table, we have created a(n):
   1. transitive dependency.
   2. insertion anomaly.
   3. referential integrity constraint.
   4. normal form.
4. A key:
   1. must always be composed of two or more columns.
   2. can only be one column.
   3. identifies a row.
   4. identifies a column.
5. SQL views can be used to hide:
   1. columns and rows only.
   2. complicated SQL syntax only.
   3. both of the above can be hidden by an SQL view.
   4. None of the above is correct
6. The SQL ALTER statement can be used to:
   1. change the table structure.
   2. change the table data.
   3. add rows to the table.
   4. delete rows from the table.
7. In a 1:N relationship, the foreign key is placed in:
   1. either table without specifying parent and child tables.
   2. the parent table.
   3. the child table.
   4. either the parent table or the child table.
8. A primary key should be defined as:
   1. NULL.
   2. NOT NULL.
   3. Either of the above can be used.
   4. None of the above are correct.
9. A UNION query is which of the following?
   1. Combines the output from no more than two queries and must include the same number of columns.
   2. Combines the output from no more than two queries and does not include the same number of columns.
   3. Combines the output from multiple queries and must include the same number of columns.
   4. Combines the output from multiple queries and does not include the same number of columns.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |