Autoevaluación UT5.-Programación Bases de Datos.

1. Selecciona la afirmación correcta:
   1. Los procedimientos almacenados son útiles cuando múltiples aplicaciones cliente se escriben en distintos lenguajes o funcionan en distintas plataformas, pero necesitan realizar la misma operación en la base de datos.
   2. Los procedimientos almacenados pueden mejorar el rendimiento ya que se necesita enviar menos información entre el servidor y el cliente.
   3. Los procedimientos almacenados le permiten tener bibliotecas o funciones en el servidor de base de datos.
   4. **Todas las anteriores.**
   5. Ninguna de las anteriores.
2. Seleccione la/s afirmación/es correcta/s:
   1. **La cláusula RETURNS puede especificarse sólo con FUNCTION, donde es obligatorio.**
   2. Cuando se borra una base de datos, todos los procedimientos almacenados asociados con ella NO se borran.
   3. **Los procedimientos almacenados requieren la tabla proc en la base de datos MySQL.**
   4. **Un procedimiento se invoca usando un comando CALL, y sólo puede pasar valores usando variables de salida.**
3. Los procedimientos almacenados pueden utilizar LOAD DATA INFILE.
   1. Verdadero.
   2. **Falso.**
4. Cambiar el delimitador por otro valor del “;” antes de la creación de un procedimiento almacenado sirve para pasar el delimitador ; usado en el cuerpo del procedimiento a través del servidor en lugar de ser interpretado por el mismo .
   1. **Verdadero.**
   2. Falso.
5. El comando para ver la definición de un procedimiento almacenado es:
   1. SHOW PROCEDURE <nombre\_procedimiento>
   2. **SHOW CREATE PROCEDURE <nombre\_procedimiento>**
   3. SHOW <nombre\_procedimiento>
   4. Ninguna de las anteriores.
6. La sentencia SELECT id,data INTO x,y FROM test.t1 LIMIT 1; :
   1. Introduce en las variables id y data los valores de los atributos x e y resultados de la consulta.
   2. **Introduce en las variables x e y los valores de los atributos id y data resultado de la consulta.**
   3. Las dos anteriores.
   4. Ninguna de las anteriores.
7. Para el manejo de errores se utiliza:
   1. Cursores
   2. **Handlers**
   3. Triggers.
   4. Ninguno de los anteriores.
8. Los cursores deben declararse después de los handlers:
   1. Verdadero.
   2. **Falso**.
9. La forma de abandonar cualquier control de flujo etiquetado en un procedimiento almacenado es con:
   1. LOOP.
   2. **LEAVE**.
   3. ITERATE.
   4. Ninguna de las anteriores.
10. Un disparador (TRIGGER) es un objeto con nombre en una base de datos que se asocia con una tabla, y se activa cuando ocurre un evento en particular para esa tabla.
    1. **Verdadero**.
    2. Falso.
11. Las sentencias que pueden activar el disparador (TRIGGER) son:
    1. CREATE, UPDATE e INSERT
    2. **INSERT, UPDATE y DELETE.**
    3. INSERT, UPDATE y DROP.
    4. Ninguna de las anteriores.
12. El disparador (TRIGGER) puede activarse antes (BEFORE) ó después (AFTER) de la sentencia que lo activa.
    1. **Verdadero**.
    2. Falso.
13. Los cursores nos permiten almacenar un conjunto de filas de una tabla en una estructura de datos que podemos ir recorriendo de forma secuencial.
    1. **Verdadero**
    2. Falso.
14. La operación que podemos utilizar para ir obteniendo las filas de un cursor abierto es:
    1. DECLARE
    2. CLOSE
    3. **FETCH**
    4. Ninguna de las anteriores.
15. Las secuencias de operaciones a utilizar con **cursores** son:
    1. OPEN, DECLARE, FETCH, CLOSE
    2. **DECLARE, OPEN, FETCH, CLOSE**
    3. FETCH, DECLARE, OPEN, CLOSE
    4. Ninguna de los anteriores.
16. Las columnas de la tabla asociada con el disparador pueden referenciarse empleando los alias OLD y NEW. Para hacer **referencia a una columna de una fila existente**, antes de ser actualizada o borrada utilizaríamos:
    1. **OLD.nombre\_col**
    2. NEW.nombre\_col
    3. Es indiferente.
    4. Ninguna de las anteriores.
17. El uso de SET NEW.nombre\_col = valor y SET nombre\_var = NEW.nombre\_col necesita que se tengan los siguientes privilegios respectivamente:
    1. **UPDATE y SELECT sobre la columna.**
    2. SELECT y UPDATE sobre la columna.
    3. SELECT y DELETE sobre la columna.
    4. Ninguna de las anteriores.
18. Para que un handler continúe la rutina actual tras la ejecución del comando handler, el tipo en la declaración debería ser:
    1. EXIT
    2. **CONTINUE**
    3. UNDO
    4. Ninguna de las anteriores.
19. La sintaxis para declarar un cursor es la siguiente: DECLARE nombre\_cursor CURSOR FOR sentencia\_select. Indica las afirmaciones falsas:
    1. **El comando SELECT puede tener la cláusula INTO.**
    2. El comando SELECT no puede tener la cláusula INTO.
    3. Pueden definirse varios cursores en una rutina, pero cada cursor en un bloque debe tener un nombre único.
20. Indica la afirmación correcta. Los constructores de control de flujo son:
    1. **IF, CASE, LOOP, WHILE, ITERATE y LEAVE. Los bucles FOR no están soportados.**
    2. IF, CASE, LOOP, WHILE, ITERATE y LEAVE. Los bucles FOR también están soportados.
    3. ITERATE sólo puede aparecer en comandos LOOP.
    4. Ninguna de las anteriores.