**Score: 60.0% (6/10)**

x1. Puede usar el operador BETWEEN para cuál de los siguientes escenarios

*  A. encontrar todas las filas entre 2 valores con tipo DATETIME
*  B. encontrar todas las filas entre 2 valores con tipo VARCHAR
*  C. encontrar todas las filas entre 2 valores con tipo INT
*  D. Ambos A y C
*  E. Todo lo anterior.

x2. 'ORDER BY id DESC' mostrará primero el registro más reciente (esto supone que hemos creado una tabla usando las convenciones adecuadas con id como clave principal única)

*  Verdadero
*  Falso

3. Dónde aplicamos las funciones mysql?

*  cualquiera de las columnas especificadas en la instrucción select
*  en la cláusula where
*  cualquier lugar dentro de una consulta donde haga referencia a una columna
*  solo cuando se usa un grupo por declaración

4. En el query 'SELECT \* FROM users LEFT JOIN posts ON users.id = posts.user\_id;'  la tabla izquierda es post

*  Verdadero
*  Falso

5. Qué esperarias que se devuelva de la siguiente consulta? 'SELECT \* FROM customers JOIN orders ON customers.id = orders.customer\_id;'

*  Solo los pedidos donde el customer\_id tiene una coincidencia para la identificación en la tabla de clientes
*  Los pedidos donde el customer\_id tiene una coincidencia para la identificación en la tabla de clientes y todos los clientes que no tienen una coincidencia
*  Los pedidos donde el customer\_id no tiene una coincidencia para el ID en la tabla de clientes y todos los pedidos que no tienen una coincidencia
*  Todos los registros en ambas tablas.

6. Es aconsejable especificar una función de agrupación cuando se utiliza una instrucción GROUP BY

*  Verdadero
*  Falso

x7. Cuál sería el resultado de la siguiente consulta? 'SELECT DATE\_FORMAT (' 2012-03-09 12:30 ','% W% M% D% Y at% r ') AS great\_time';

*  Viernes Marzo 9th 2012 at 12:30:00 PM
*  Vie Mar 9 12' at 12:30
*  Marzo 9th at 12:30:00
*  Viernes Mar 9 2012 at 12:30 PM

8. En la siguiente consulta, ¿qué esperaría que se devuelva? SELECCIONE clients.company, SUM (billing.amount) AS total\_sales FROM clients LEFT JOIN billing ON billing.client\_id = client.id GROUP BY clients.company HAVING SUM (billing.amount )> 3500;

*  Una lista de todas las empresas y sus importes totales de facturación.
*  Una lista de cada compañía cuyos montos totales de facturación son superiores a 3500
*  Una lista de empresas con sus ventas totales.
*  Una lista de todos los importes de facturación superiores a 3500

9. Cuándo desearía utilizar una IZQUIERDA IZQUIERDA en lugar de una UNIÓN?

*  cuando hay registros que desea mostrar en la tabla de la izquierda que no coinciden con los registros de la tabla de la derecha
*  cuando hay registros que desea mostrar en la tabla derecha que no coinciden con los registros de la tabla izquierda
*  si desea que se muestren todos los registros de ambas tablas, incluso si no tienen una coincidencia
*  Siempre puedes usar LEFT JOIN en lugar de JOIN

x10. Qué registros se mostrarán en la siguiente consulta 'SELECCIONAR ID, nombre\_de los usuarios LIMIT 11,10;'

*  10 registros de nombre e identificación del usuario, comenzando con el 12º registro en el resultado (independientemente de la identificación de ese registro)
*  2 registros de nombre e identificación del usuario, específicamente los usuarios con los identificadores 11 y 10
*  10 registros de nombre e identificación del usuario, comenzando con id = 11 (independientemente de si se encuentran identificadores inferiores en la tabla)
*  11 registros de nombre e identificación del usuario, comenzando con el décimo registro en el resultado (independientemente de la identificación de ese registro)