

Este documento era antes de SumaCRM (ahora Efficacy)
[Clic para probar efficacy Gratis](#)

Prueba de programación

Todos los puntos descritos debajo suponen un servidor LAMP dedicado. Y el uso de javascript (puro o usando librerías) si ves que es necesario. El objetivo es suponer como si estuvieras ya trabajando con nosotros..

Tienes 2 horas para hacerlo. El objetivo es ver cómo gestionarías tu tiempo en estas 2 horas.

Por favor detallar lo más posible en tus respuestas.

Punto 1:

Descripción del entorno

- Base de Datos: MySQL InnoDB.
- Existe un CRM. Existen N usuarios en una cuenta determinada.
- Entidades presentes
 - Contactos, notas y comentarios.
 - Una nota tiene si o si un ID_CONTACTO asociado.
 - Un comentario tiene si o si un ID_NOTA asociado.
 - En otras palabras, una nota siempre será asociada a un contacto determinado.
Y si existe un comentario siempre será asociado a una Nota determinada.
- La privacidad/visibilidad de estas entidades, entre los usuarios de la cuenta, es un tema de suma importancia para el CRM.
- Un usuario puede definir lo siguiente, relacionado a la privacidad/visibilidad:
 - Solo para mi
 - Para mi y los siguientes (N) usuarios
 - Abierto (visible para todos)
- Suponer que existen las siguientes tablas:
 - Sumacrm_contactos
 - 18.000.000 registros
 - Campos: id, id_account, firstname, lastname, email, telephone, date_added
 - Sumacrm_notas
 - 5.000.000 registros
 - Sumacrm_comentarios.
 - 2.500.000 registros
- La visibilidad/privacidad sobre los contactos, las notas y los comentarios puede ser cambiada en cualquier momento.

Ejemplos:

- Un usuario puede definir esa privacidad X a un Contacto. Esto hará que el Contacto sólo sea visible para quienes estén autorizados. Esto se debe extender a la

privacidad

Este documento era antes de SumaCRM (ahora Efficcy)

[Clic para probar efficacy Gratis](#)

de todas las notas y comentarios que un Contacto tenga, salvo aquellos que tengan propia visibilidad, que debería respetarse.

- Un usuario puede definir privacidad “Abierto” a un Contacto. Y definir otros niveles de privacidad a las notas del Contacto. Hará que el Contacto sea visible para todos los usuario de la cuenta, y solo podrá ver las notas a la cual tenga permiso de visibilidad.

Indica y define:

- la estructura de datos (definición de estructura en MySQL) que elegirías para almacenar información relacionada a la visibilidad/privacidad.
 - Por favor da tus razones de la estructura elegida.
 - Que tablas usarías?
 - Que campos definirías? Con que índices?
 - Puedes incluir un diagrama de relación para explicar mejor la estructura elegida.
- Dá ejemplo de sentencias MySQL que son necesarios, usando tu estructura definida, para poder filtrar, respetando la privacidad (excluyendo lo que un usuario NO puede ver)
 - Listado de contactos que el usuario que está navegando puede ver
 - Listado de notas que el usuario que está navegando puede ver
 - Listado de comentarios que el usuario que está navegando puede ver
 - Listado de notas de un contacto que el usuario que está navegando puede ver
 - Listado de notas de un contacto creadas por otro usuario, que el usuario que está navegando puede ver.

Punto 2

Utilizando la estructura que definiste en el Punto 1, escribir el código PHP, orientado a objetos, que resuelva lo siguiente: Importación de contactos a la base de datos de la manera más rápida.

Puntos a considerar:

- La cuenta tiene 1800000 contactos.
- Parte de los contactos, tiene algún tipo de privacidad (mismas definidas en el Punto 1)
- La importación debe respetar la privacidad de contactos.
- Qué harían al tratar de importar filas de contactos desde el CSV, que coinciden con contactos que están presentes en la agenda de contactos, pero yo usuario, no tengo visibilidad permitida a dicho contacto.
 - No está permitido modificar un contacto existente si no tengo permiso de visibilidad al contacto.

CSV

- Variable de entrada: ruta de un fichero CSV ya alojado en el servidor.
- Se trata de importar un CSV con 1.000.000 filas. Cada fila es un contacto.

- En el CSV existen filas correspondientes a contactos que ya existen en la agenda de contactos de suma, y el dato unívoco es la dirección de email.
- En el CSV existen contactos sin dirección de email.
- En el caso de contactos existentes (mismo email), debe actualizar información del contacto pisando con información que viene en el CSV.
- No se permiten contactos duplicados (con mismo email)

Este documento era antes de SumaCRM (ahora Efficacy)

[Clic para probar efficacy Gratis](#)