

Explicación detallada de las consultas e inserciones SQL en la aplicación de chat

Ejemplo de flujo de datos

1. Usuario envía mensaje de texto:

- Se inserta un registro en MENSAJE (con metadatos y posible referencia a mensaje citado).
- Se inserta el texto en CONTENIDO con IDTIPOCONTENIDO = '2'.

2. Usuario envía archivo:

- Se inserta un registro en MENSAJE.
- Se inserta la ruta del archivo en CONTENIDO con IDTIPOCONTENIDO = '1' y el tipo de archivo.

3. Usuario responde a un mensaje:

- El nuevo mensaje en MENSAJE incluye los campos MEN_USU_CONSECUSER, MEN_CONSECUSER, MEN_CONSMENSAJE apuntando al mensaje original.

1. Registro de Usuario

Consulta:

```
INSERT INTO USUARIO (  
  CONSECUSER, NOMBRE, APELLIDO, "USER",  
  PASSWORD, EMAIL, CELULAR, FECHAREGISTRO  
)  
VALUES (  
  :id, :nombre, :apellido,  
  :username, :password,  
  :email, :celular, SYSDATE  
)
```

Explicación:

- Inserta un nuevo usuario en la tabla USUARIO.
- CONSECUSER: identificador único (UUID) para cada usuario.
- NOMBRE, APELLIDO, "USER", PASSWORD, EMAIL, CELULAR: datos personales del usuario.
- FECHAREGISTRO: almacena la fecha y hora de registro usando SYSDATE.

2. Login de Usuario

Consulta:

```
SELECT *  
  FROM USUARIO  
 WHERE "USER" = :username  
    AND PASSWORD = :password
```

Explicación:

Busca un usuario por nombre y contraseña; si existe, permite iniciar sesión.

3. Agregar Amigo

Consultas:

1. SELECT 1 FROM USUARIO WHERE CONSECUSER = :id SELECT 1 FROM USUARIO WHERE "USER" = :amigoUser
2. SELECT CONSECUSER FROM USUARIO WHERE "USER" = :amigoUser
3. SELECT 1 FROM AMIG_ WHERE (CONSECUSER = :a AND USU_CONSECUSER = :b) OR (CONSECUSER = :b AND USU_CONSECUSER = :a)
4. INSERT INTO AMIG_ (CONSECUSER, USU_CONSECUSER) VALUES (:a, :b)

Explicación:

Valida existencia de usuarios y no duplicar la amistad; luego inserta la relación en AMIG_.

4. Enviar Mensaje de Texto (y/o Cita)

Consultas:

1. SELECT NVL(MAX(CONSMENSAJE),0)+1 AS NEXTMSG FROM MENSAJE WHERE USU_CONSECUSER = :rem AND CONSECUSER = :dest

```

2. INSERT INTO MENSAJE (
    USU_CONSECUSER, CONSECUSER, CONSMENSAJE,
    FECHAREGMEN, MEN_USU_CONSECUSER, MEN_CONSECUSER,
    MEN_CONSMENSAJE
)
VALUES (
    :rem, :dest, :cons, SYSDATE,
    :men_usu_consecuser, :men_consecuser, :men_consMensaje
)

INSERT INTO CONTENIDO (
    USU_CONSECUSER, CONSECUSER, CONSMENSAJE,
    CONSECONTENIDO, IDTIPOARCHIVO,
    IDTIPOCONTENIDO, LOCALIZACONTENIDO
)
VALUES (
    :rem, :dest, :cons, 1, NULL, '2', :texto
)

```

Explicación:

Genera un consecutivo para la conversación, inserta metadatos y luego el texto.

5. Enviar Archivo

Consultas:

```

3. Igual al paso 1 de texto.

2. Igual al paso 2 (sin hilo).

3. INSERT INTO CONTENIDO (
    USU_CONSECUSER, CONSECUSER, CONSMENSAJE,
    CONSECONTENIDO, IDTIPOARCHIVO,
    IDTIPOCONTENIDO, LOCALIZACONTENIDO
)
VALUES (
    :rem, :dest, :cons, 1, :tipoArchivo, '1', :ruta
)

```

Explicación:

Guarda la ruta del archivo y su tipo.

6. Obtener Mensajes entre Usuarios

Consulta:

```

SELECT * FROM (
  SELECT
    m.USU_CONSECUSER, m.CONSECUSER, m.CONSMENSAJE,
    m.FECHAREGMEN, c.LOCALIZACONTENIDO,
    c.IDTIPOCONTENIDO, c.IDTIPOARCHIVO,
    m.MEN_USU_CONSECUSER, m.MEN_CONSECUSER,
    m.MEN_CONSMENSAJE,
    ROW_NUMBER() OVER (ORDER BY m.FECHAREGMEN DESC) AS RN
  FROM MENSAJE m
  JOIN CONTENIDO c
    ON m.USU_CONSECUSER = c.USU_CONSECUSER
   AND m.CONSECUSER      = c.CONSECUSER
   AND m.CONSMENSAJE     = c.CONSMENSAJE
  WHERE ((m.USU_CONSECUSER = :u1 AND m.CONSECUSER = :u2)
        OR (m.USU_CONSECUSER = :u2 AND m.CONSECUSER = :u1))
        AND (c.IDTIPOCONTENIDO = '1' OR c.IDTIPOCONTENIDO = '2'))
) WHERE RN BETWEEN :startRow AND :endRow
ORDER BY RN

```

Explicación:

Recupera mensajes y archivos entre dos usuarios, paginados y con hilos.

7. Descargar Archivo

No es SQL puro; el endpoint sirve el archivo usando la ruta en LOCALIZACONTENIDO.

Relaciones clave de la BD:

- USUARIO: datos de los usuarios.
- AMIG_: pares de amigos.
- MENSAJE: metadatos de mensajes e hilos.
- CONTENIDO: cuerpo de texto o referencia a archivos.