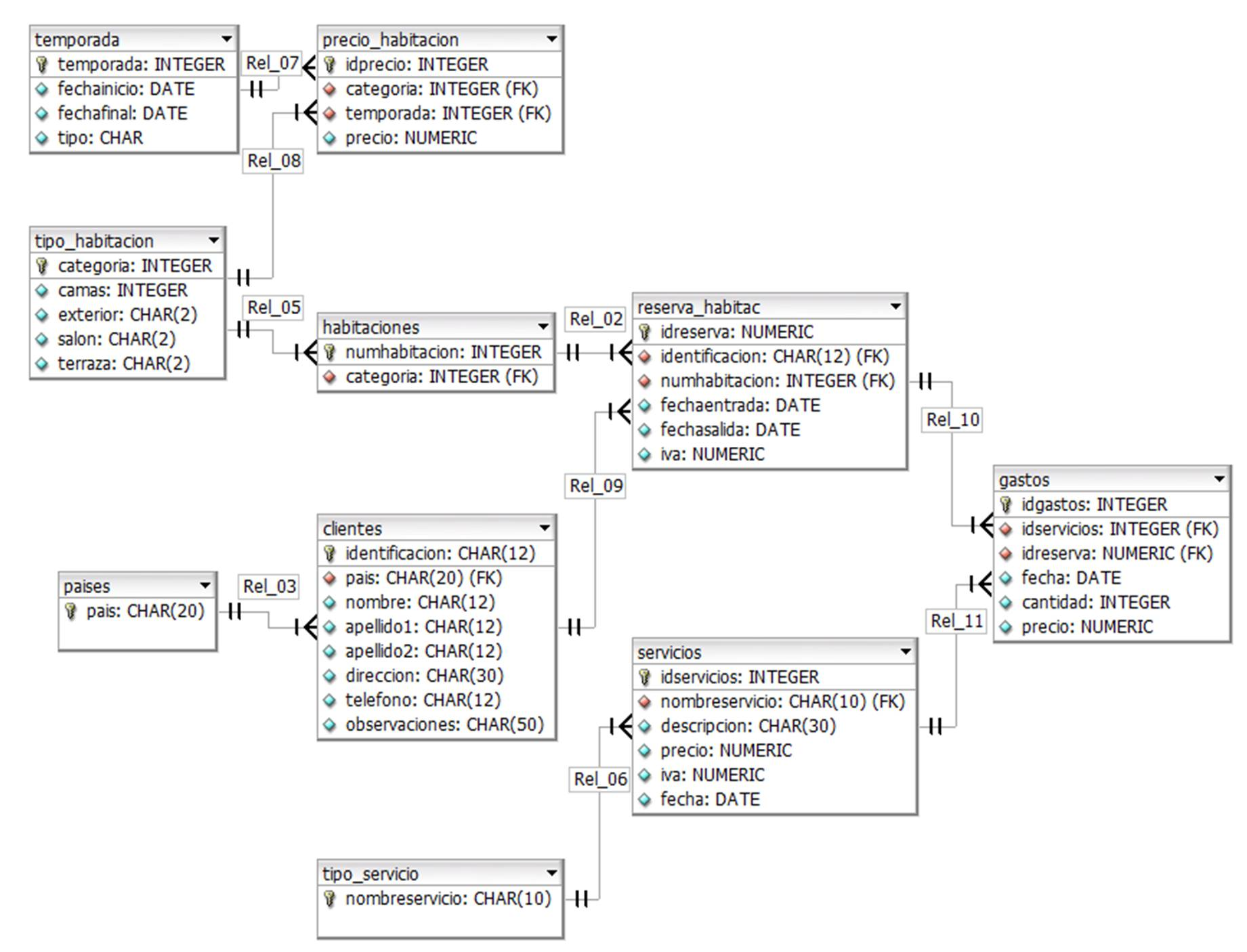
## Administración de Base de datos

**Nombre: Aciertos: /29 Calif:**

# Usar la BD mostrada en la imagen



1. Crear un procedimiento llamado **reserva** con las siguientes características.
2. El procedimiento recibe como parámetros de **entrada:** idreserva, nombre de cliente, numhabitacion, fechaentrada
3. Mostrará un mensaje con el tipo de error que detecte, o un mensaje de éxito en otro caso **( 3 aciertos)**

El procedimiento realiza los siguientes pasos:

1. Inicia una transacción. **(1 acierto)**
2. Busca la clave del cliente de acuerdo con el nombre recibido como entrada **(1 acierto)**
3. Calcula la fecha de salida, la cual serán 3 días después de la fecha de entrada, usando la función de SQL:

## DATE\_ADD(fecha ,INTERVAL numdias DAY); (1 acierto)

1. Inserta un registro en la tabla reserva\_habitac poniendo como IVA el valor: 0.16 **( 2 aciertos)**
2. Inserta un registro también en gastos con los valores: **(3 aciertos)**
   * Calcula a través de una **función** el número de gasto que sigue **(3 aciertos – 1 acierto de llamada)**

o Con el servicio **“COMEDOR”**, para esto con otro **procedimiento** recupera el idservicio y el precio **(3 aciertos– 1 acierto de llamada)**

* + La fecha del gasto corresponde a un día después de que se hace la reservación
  + La cantidad será 2
  + El id de reserva se toma del recibido como parámetro inicialmente

1. Comprueba si ha ocurrido algún error en las operaciones anteriores. Si no ocurre ningún error entonces aplica un COMMIT a la transacción y si ha ocurrido algún error aplica un ROLLBACK.

Deberá manejar los siguientes errores que puedan ocurrir durante el proceso.

* ERROR 1048 (El cliente no existe)
* ERROR 1062 (Llave primaria duplicada)

# Probar con los siguientes valores e indicar al lado el resultado mostrado en tu procedimiento: **(4 aciertos)**

call reserva(1,”Felipe”, 106, ‘2022/07/15’); call reserva(10,”Felipe”, 106, ‘2022/07/15’); call reserva(12,"Juan", 106, ‘2022/07/15’); call reserva(15,"Luis", 101, ‘2022/07/10’);

1. **Agregar una tabla** que se llame **pagos** que tenga las siguientes columnas:

* id (entero sin signo, autoincremento y clave primaria)
* fecha\_hora: marca de tiempo con el instante del pago (fecha y hora)
* monto: el valor del pago (real)
* idgastos: clave de la reservación que se está pagando (numeric)

1. Escribir un **trigger** que nos permita llevar un control de los pagos que van realizando los clientes. Los detalles de implementación son los siguientes: **(5a)**
   * Nombre: trigger\_pago
   * Se ejecuta sobre la tabla gastos.
   * Se ejecuta *después* de hacer la inserción de un gasto.
   * Cada vez que un cliente contrate un servicio (es decir, se hace una inserción en la tabla gastos), el *trigger* deberá insertar un nuevo registro en una tabla llamada pagos, para esto se debe multiplicar el precio por la cantidad del gasto y ese será el monto del pago

# Probarlo con los siguientes Insert y tomar captura de lo que contiene después de los insert la tabla pagos

INSERT INTO gastos(idgastos, idRESERVA, idSERVICIOS, Fecha, Cantidad, Precio) VALUES(10, 5, 2, '2022-08-11 11:30', 3, 25);

INSERT INTO gastos(idgastos, idRESERVA, idSERVICIOS, Fecha, Cantidad, Precio) VALUES(11, 7, 3, '2022-08-08 11:30', 10, 12);

1. **Modificar la tabla** clientes agregando un campo llamado correo de tipo varchar(50).
2. Escribir un **procedimiento** llamado **crear\_correo** que dados los parámetros de entrada: nombre, apellido1, apellido2, cree una dirección de correo y la devuelva como salida. **(3a)**

* Procedimiento: crear\_correo
* Entrada:
  + nombre (cadena de caracteres)
  + apellido1 (cadena de caracteres)
  + apellido2 (cadena de caracteres)
* Salida:
  + correo (cadena de caracteres)

La dirección de correo electrónico devuelta tendrá el siguiente formato:

* El primer carácter del parámetro nombre.
* Los tres primeros caracteres del parámetro apellido1.
* Los tres primeros caracteres del parámetro apellido2.
* El carácter @.
* El dominio: gmail.com pasado como parámetro.

**TIP:** usar las funciones de SQL: **CONCAT** (cadenas separadas por comas de lo que se quiere concatenar) y

**LEFT** (cadena, num caracteres a extraer)

1. Escriba un **procedimiento** llamado **actualizar\_correos** que permita crear un email para todos los clientes que ya existen en la tabla. **(12a)**

Procedimiento: **actualizar\_correos**

Entrada: no hay

Proceso: Usar un cursor para que por cada registro de la tabla clientes se envíen los datos requeridos al procedimiento crear\_correo, Posteriormente se debe actualizar el correo en la tabla clientes.

Manejo de errores: usarlo para controlar el cursor

# Llamar el procedimiento, y tomar captura de lo que contiene la tabla clientes después