

PROYECTO: FULL DAY

SCAPE

NOMBRE: GIULLIANA VERGARA

COMISION: 47360

INDICE

- **TEMATICA**
- **LISTADO DE TABLAS**
- **DIAGRAMA ENTIDAD RELACION**
- **LISTADO DE VISTAS (VIEWS)**
- **LISTADO DE FUNCIONES**
- **LISTADO DE STORE PROCEDURE**
- **LISTADO DE LOG**
- **LISTADO DE TRIGGERS**

"FULL DAY SCAPE"

La base de datos **"FULL DAY SCAPE"** es un sistema integral de gestión de reservas y promociones para un establecimiento de entretenimiento. Esta base de datos se compone de cinco tablas principales: **"USUARIOS"**, **"RESERVAS"**, **"PAQUETES"**, **"PROMOCIONES"** y **"SERVICIOS"**. Cada una de estas tablas desempeña un papel fundamental en la organización y administración de las actividades y servicios ofrecidos por **"FULL DAY SCAPE"**.

La tabla **"USUARIOS"** almacena información detallada sobre los clientes y usuarios del establecimiento. Esta tabla es esencial para llevar un registro de quienes hacen uso de los servicios y para personalizar la experiencia de los clientes.

La tabla **"RESERVAS"** está diseñada para gestionar las reservas de los clientes, proporcionando detalles sobre fechas, horarios y otros datos relevantes. Además, esta tabla establece relaciones uno a uno tanto con la tabla **"USUARIOS"** como con la tabla **"PAQUETES"**, permitiendo una conexión directa entre los clientes y los servicios reservados.

La tabla **"PAQUETES"** enumera los diferentes tipos de servicios o actividades que los clientes pueden adquirir. Estos paquetes pueden incluir una variedad de servicios o actividades recreativas.

La tabla **"PROMOCIONES"** almacena información sobre las ofertas y promociones especiales disponibles para los clientes. Estas promociones pueden estar vinculadas a uno o varios paquetes, lo que se gestiona a través de relaciones entre las tablas.

La tabla **"SERVICIOS"** enumera todos los servicios individuales que pueden estar disponibles en el establecimiento, como masajes, restaurantes, o actividades específicas. Estos servicios pueden ser parte de uno o varios paquetes.

Además, hay dos tablas intermedias que actúan como puentes para las relaciones entre los paquetes y las promociones, así como entre los paquetes y los servicios. Estas relaciones permiten asociar múltiples paquetes a promociones específicas o vincular paquetes a diversos servicios disponibles.

también agregando la tabla **"LOG_PAQUETES"** y **"LOG_RESERVAS"** tienen como objetivo principal llevar un registro detallado de las operaciones relacionadas con paquetes y reservas en una base de datos. Estas tablas permiten rastrear quién realizó una operación, cuándo se llevó a cabo y proporcionan una descripción de la misma. La información almacenada en estas tablas de registro es valiosa para el seguimiento de cambios en paquetes y reservas a lo largo del tiempo y para fines de auditoría. Esto facilita la identificación de modificaciones específicas, tanto antes como después de una actualización, lo que puede ser útil para la detección de errores, seguimiento de cambios y garantía de integridad de datos en el sistema. En resumen, estas tablas ayudan a mantener un registro histórico y transparente de las operaciones críticas en la base de datos relacionadas con paquetes y reservas.

En resumen, la base de datos **"FULL DAY SCAPE"** es una herramienta esencial para la gestión integral de un establecimiento de entretenimiento, que permite a los usuarios realizar reservas, acceder a paquetes personalizados, beneficiarse de promociones especiales y disfrutar de una amplia variedad de servicios.

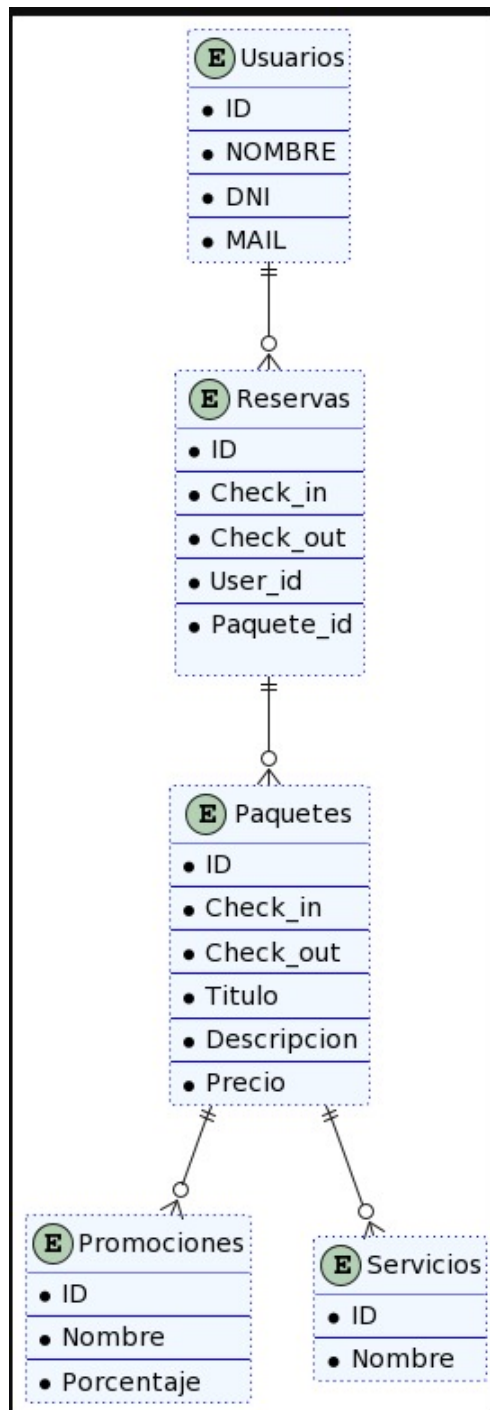
LISTADO DE TABLAS:

TABLA	NOMBRE ABREVIADO	NOMBRE DEL CAMPO COMPLETO	CLAVE PRIMARIA (PK)	CLAVE FORANEA (FK)	TIPO DE DATOS
USUARIOS	ID	IDENTIFICADOR DEL USUARIO	PK		INT
	NOMBRE	NOMBRE DEL USUARIO			VARCHAR
	MAIL	MAIL DEL USUARIO			VARCHAR
	DNI	DNI DEL USUARIO			INT
RESERVAS	ID	IDENTIFICADOR DE LA RESERVA	PK		INT
	PRECIO	PRECIO DE LA RESERVA			FLOAT
	CHECK_IN	HORA DE ENTRADA			DATETIME
	CHECK_OUT	HORA DE SALIDA			DATETIME
	USER_ID	IDENTIFICADOR DEL USUARIO QUE REALIZO LA RESERVA		FK	INT
	PAQUETE_ID	IDENTIFICADOR DEL PAQUETE DE LA RESERVA		FK	INT
PAQUETES	ID	IDENTIFICADOR DEL PAQUETE	PK		INT
	TITULO	TITULO DEL PAQUETE			VARCHAR
	CHECK_IN	HORA DE ENTRADA			DATETIME
	CHECK_OUT	HORA DE SALIDA			DATETIME
	DESCRIPCION	DESCRIPCION DEL PAQUETE			VARCHAR
	PRECIO	PRECIO DEL PAQUETE			FLOAT
	ID	IDENTIFICADOR DE LA PROMOCION	PK		INT

PROMOCIONES	NOMBRE	NOMBRE DE LA PROMOCION			VARCHAR
	PORCENTAJE	PORCENTAJE DE LA PROMOCION			FLOAT
SERVICIOS	ID	IDENTIFICADOR DEL SERVICIO	PK		INT
	NOMBRE	NOMBRE DEL SERVICIO			VARCHAR
PAQUETES_PRO MOCIONES	ID	IDENTIFICADOR DE LA RELACION DEL PAQUETE Y LA PROMOCION	PK		INT
	PAQUETE_ID	IDENTIFICADOR DEL PAQUETE		FK	INT
	PROMOCIONES_ID	IDENTIFICADOR DE LAS PROMOCIONES		FK	INT
PAQUETES_SER VICIOS	ID	IDENTIFICADOR DE LA RELACION DEL PAQUETE Y LOS SERVICIOS	PK		INT
	PAQUETE_ID	IDENTIFICADOR DEL PAQUETE		FK	INT
	SERVICIOS_ID	IDENTIFICADOR DE LOS SERVICIOS		FK	INT
	ID_LOG_PAQUETES	IDENTIFICADOR DE LOG DE PAQUETES	PK		INT
	USUARIO_REALIZADOR	USUARIO LOGUEADO EN LA BASE			VARCHAR
LOG_PAQUETES	TIPO_OPERACION	ACCION REALIZADA EN LA BASE			VARCHAR
	FECHA_HORA	FECHA Y HORA CUANDO SE REALIZO LA OPERACION			DATETIME
	DESCRIPCION	VALORES DEL REGISTRO DE TABLA			VARCHAR
	ID_PAQUETE	IDENTIFICADOR DEL PAQUETE		FK	INT

	ID_LOG_RESE RVA	IDENTIFICADOR DE LOG DE RESERVA	PK		INT
	USUARIO_REAL IZADOR	USUARIO LOGUEADO EN LA BASE			VARCHAR
LOG_RESERVAS	TIPO_OPERACI ON	ACCION REALIZADA EN LA BASE			INT
	FECHA_HORA	FECHA Y HORA CUANDO SE REALIZO LA OPERACION			DATETIME
	DESCRIPCION	VALORES DEL REGISTRO DE TABLA			VARCHAR
	ID_RESERVA	IDENTIFICADOR DE LA RERSERVA		FK	INT

DIAGRAMA ENTIDAD RELACION



LISTADO DE VISTAS (VIEWS)

- **"usuarios_con_reserva"** muestra información de usuarios que han realizado reservas, incluyendo el precio, nombre del usuario, correo electrónico, nombre del paquete y fecha de ingreso.
- **"paquetes_con_promociones"** muestra detalles de los paquetes que tienen promociones, incluyendo el título del paquete, fecha de ingreso, precio, nombre de la promoción y el porcentaje de descuento.
- **"paquetes_permiten_mascotas"** se enfoca en paquetes que permiten mascotas y muestra información sobre el título del paquete, fecha de ingreso, precio y el nombre del servicio asociado a las mascotas.
- **"paquetes_comida_transporte"** muestra paquetes que incluyen servicios de comida o transporte, proporcionando información sobre el título del paquete, fecha de ingreso, precio y el nombre del servicio correspondiente.
- **"paquetes_baratos"** se centra en los paquetes que tienen un precio menor a 30 unidades monetarias, mostrando solo el título y el precio de esos paquetes.

Estas vistas permiten acceder de manera más sencilla a información específica en la base de datos relacionada con paquetes, reservas, servicios, promociones y precios.

LISTADO DE FUNCIONES

- **Primera función:** buscar_paquete_por_titulo

Esta función toma como entrada el título de un paquete (titulo_paquete) y devuelve una cadena de caracteres (VARCHAR(255)) que contiene información sobre el paquete con el título proporcionado, incluyendo el título y el precio del paquete.

La función consulta la tabla "paquetes" y busca un paquete con el título que coincide con el parámetro proporcionado.

La información se concatena en una cadena y se almacena en la variable resultado. La función retorna esta información.

- **Segunda función:** buscar_paquetes_por_servicio

Esta función toma como entrada el nombre de un servicio (servicio_nombre) y devuelve una cadena de caracteres (VARCHAR(2000)) que contiene una lista de títulos de paquetes que incluyen ese servicio.

La función consulta las tablas "paquetes," "paquetes_servicios," y "servicios" para encontrar todos los paquetes que ofrecen el servicio especificado.

Los títulos de los paquetes se concatenan en una cadena utilizando la función GROUP_CONCAT y se almacenan en la variable resultado.

La función retorna esta lista de títulos de paquetes.

En resumen, estas dos funciones permiten buscar información específica sobre paquetes basada en el título del paquete o en un servicio en particular, proporcionando resultados útiles y legibles para el usuario.

LISTADO DE STORE PROCEDURE

- **Primer Stored Procedure:** ModificarDescuentoPromocion

Este procedimiento recibe dos parámetros de entrada: id_promocion (un entero) y porcentaje_descuento (un número decimal).

Verifica si una promoción con el id_promocion proporcionado existe en la tabla "promociones."

Si la promoción existe, actualiza el porcentaje de descuento de esa promoción con el valor proporcionado.

Luego, asigna un mensaje de éxito si la actualización se realizó con éxito, o un mensaje de error si la promoción no se encontró.

El procedimiento devuelve un mensaje como resultado.

- **Segundo Stored Procedure:** AplicarDescuentoReserva

Este procedimiento también recibe dos parámetros de entrada: id_reserva (un entero) y descuento (un número decimal).

Verifica si una reserva con el id_reserva proporcionado existe en la tabla "reservas."

Si la reserva existe, aplica un descuento al precio de la reserva restando el valor del descuento.

Luego, asigna un mensaje de éxito si la actualización se realizó con éxito, o un mensaje de error si la reserva no se encontró.

El procedimiento devuelve un mensaje como resultado.

En resumen, estos Stored Procedures brindan funcionalidad para modificar el porcentaje de descuento de una promoción y aplicar un descuento a una reserva, mientras proporcionan mensajes informativos sobre el resultado de la operación, ya sea de éxito o de error.

LISTADO DE LOG

- **Tabla "LOG_PAQUETES":**

id_log_paquete: Un campo numérico único y autoincremental que sirve como identificador de cada registro en esta tabla.

usuario_realizador: Almacena el nombre del usuario que realizó una operación de registro relacionada con paquetes.

id_paquete: Un número que establece una relación con la tabla "paquetes" y hace referencia al paquete al que se refiere el registro.

tipo_operacion: Un campo que registra el tipo de operación realizada, que puede ser "UPDATE_BEFORE" (antes de la actualización) o "UPDATE_AFTER" (después de la actualización).

fecha_hora: Almacena la fecha y hora en que se realizó el registro, con un valor predeterminado que es la marca de tiempo actual.

descripcion: Un campo de texto que contiene información detallada sobre la operación registrada.

FOREIGN KEY (id_paquete): Establece una relación con la tabla "paquetes" a través del campo "id_paquete."

- **Tabla "LOG_RESERVAS":**

id_log_reserva: Un campo numérico único y autoincremental que sirve como identificador de cada registro en esta tabla.

usuario_realizador: Almacena el nombre del usuario que realizó una operación de registro relacionada con reservas.

id_reserva: Un número que establece una relación con la tabla "reservas" y hace referencia a la reserva a la que se refiere el registro.

tipo_operacion: Un campo que registra el tipo de operación realizada, que puede ser "UPDATE_BEFORE" (antes de la actualización) o "UPDATE_AFTER" (después de la actualización).

fecha_hora: Almacena la fecha y hora en que se realizó el registro, con un valor predeterminado que es la marca de tiempo actual.

descripcion: Un campo de texto que contiene información detallada sobre la operación registrada.

FOREIGN KEY (id_reserva): Establece una relación con la tabla "reservas" a través del campo "id_reserva."

En resumen, estas dos tablas, "**log_paquetes**" y "**log_reservas**", se utilizan para llevar un registro detallado de las operaciones relacionadas con paquetes y reservas en la base de datos. Esto facilita el seguimiento de cambios, la auditoría y la garantía de integridad de datos en el sistema. Las tablas también mantienen relaciones con las tablas "paquetes" y "reservas" para asegurar la consistencia de los datos.

LISTADO DE TRIGGERS

Se han definido dos Triggers en SQL para la tabla "paquetes" con el propósito de registrar cambios antes y después de una actualización en la tabla.

- **Trigger BEFORE ("tr_update_paquetes_before"):**

Este trigger se ejecuta antes de que ocurra una operación de actualización en la tabla "paquetes."

Al activarse, registra los valores originales de las columnas específicas del paquete antes de la actualización en la tabla "log_paquetes."

Los datos registrados incluyen el nombre del usuario que realiza la operación, el ID del paquete, el tipo de operación ("UPDATE_BEFORE"), y una descripción que contiene los valores previos de las columnas, como el título, las fechas de check-in y check-out, y el precio.

- **Trigger AFTER ("tr_update_paquetes_after"):**

Este trigger se ejecuta después de que se haya realizado una operación de actualización en la tabla "paquetes."

Al activarse, registra los valores actualizados de las mismas columnas en la tabla "log_paquetes."

Los datos registrados incluyen el nombre del usuario que realiza la operación, el ID del paquete, el tipo de operación ("UPDATE_AFTER"), y una descripción que contiene los valores actualizados de las columnas, como el título, las fechas de check-in y check-out, y el precio.

En resumen, estos dos triggers sirven para rastrear y registrar los cambios en los datos de la tabla "paquetes" antes y después de que se realicen actualizaciones. Esto proporciona una forma de auditoría y seguimiento de los cambios, permitiendo mantener un registro de las modificaciones en los registros de paquetes.

Se han definido dos Triggers en SQL para la tabla "reservas" con el propósito de registrar cambios antes y después de una actualización en la tabla.

- **Trigger BEFORE ("tr_update_reservas_before"):**

Este trigger se ejecuta antes de que ocurra una operación de actualización en la tabla "reservas."

Al activarse, registra los valores originales de las columnas específicas de la reserva antes de la actualización en la tabla "log_reservas."

Los datos registrados incluyen el nombre del usuario que realiza la operación, el ID de la reserva, el tipo de operación ("UPDATE_BEFORE"), y una descripción que contiene los valores previos de las columnas, como el precio, las fechas de check-in y check-out.

- **Trigger AFTER ("tr_update_reservas_after"):**

Este trigger se ejecuta después de que se haya realizado una operación de actualización en la tabla "reservas."

Al activarse, registra los valores actualizados de las mismas columnas en la tabla "log_reservas."

Los datos registrados incluyen el nombre del usuario que realiza la operación, el ID de la reserva, el tipo de operación ("UPDATE_AFTER"), y una descripción que contiene los valores actualizados de las columnas, como el precio, las fechas de check-in y check-out.

En resumen, estos dos triggers sirven para rastrear y registrar los cambios en los datos de la tabla "reservas" antes y después de que se realicen actualizaciones. Esto proporciona una forma de auditoría y seguimiento de los cambios, permitiendo mantener un registro de las modificaciones en los registros de reservas.