

# **PROYECTO: FULL DAY**

## **SCAPE**

**NOMBRE: GIULLIANA VERGARA**

**COMISION: 47360**

## **INDICE**

- **INTRODUCCIÓN**
- **TEMATICA**
- **LISTADO DE TABLAS**
- **DIAGRAMA ENTIDAD RELACION**
- **LISTADO DE VISTAS (VIEWS)**
- **LISTADO DE FUNCIONES**
- **LISTADO DE STORE PROCEDURE**
- **LISTADO DE LOG**
- **LISTADO DE TRIGGERS**
- **SENTENCIAS DCL (DATA CONTROL LANGUAGE)**
- **SENTENCIAS TCL**
- **BACKUP DE BASE DE DATOS**
- **CONCLUSIÓN**

# **INTRODUCCIÓN**

La base de datos de Full Day Scape es un recurso integral para una agencia de viajes especializada en la organización y gestión de paquetes de viajes de un día completo (full days). La estructura y contenido de esta base de datos reflejan su propósito de facilitar la administración eficiente de varios aspectos clave de la agencia de viajes.

## **Componentes Clave de la Base de Datos:**

### **1. Tabla de Usuarios (usuarios):**

- Contiene información detallada de los clientes, como nombre, correo electrónico y DNI.
- Facilita la gestión personalizada de cada cliente, permitiendo a la agencia ofrecer servicios y ofertas adaptadas a sus necesidades y preferencias.

### **2. Tabla de Servicios (servicios):**

- Enumera los diferentes servicios ofrecidos en los paquetes de viaje, como Wifi, lavandería, gimnasio, transporte, etc.
- Permite a la agencia destacar la variedad y calidad de los servicios disponibles, mejorando la experiencia del cliente.

### **3. Tabla de Promociones (promociones):**

- Incluye diversas ofertas promocionales con detalles como el nombre de la promoción y el porcentaje de descuento.
- Esencial para atraer a nuevos clientes y fidelizar a los existentes mediante ofertas atractivas.

### **4. Tabla de Paquetes (paquetes):**

- Contiene los detalles de los paquetes de viaje ofrecidos, incluyendo título, fechas de check-in y check-out, descripción y precio.
- Proporciona una visión clara de las opciones de viaje disponibles, facilitando a los clientes la elección del paquete que mejor se adapte a sus intereses y presupuesto.

### **5. Tablas de Relación (paquetes\_promociones, paquetes\_servicios):**

- Establecen relaciones entre paquetes, servicios y promociones.
- Permiten personalizar paquetes con servicios y ofertas específicas, mejorando la adaptabilidad y la satisfacción del cliente.

#### **6. Tabla de Reservas (reservas):**

- Registra las reservas realizadas por los usuarios, incluyendo detalles como el precio, fechas y paquete seleccionado.
- Fundamental para el seguimiento y gestión de reservas, asegurando que se cumplan las expectativas de los clientes y se mantenga la eficiencia operativa.

#### **Funcionalidad para la Agencia de Viajes:**

**Gestión Eficiente:** La base de datos permite una administración efectiva y eficiente de todas las operaciones de la agencia, desde la reserva de paquetes hasta la aplicación de promociones.

**Personalización de Servicios:** Facilita la personalización de paquetes de viaje según las preferencias de los usuarios, lo que mejora la experiencia del cliente y aumenta la fidelidad.

**Estrategia de Marketing:** Las promociones y ofertas especiales registradas en la base de datos pueden ser utilizadas para estrategias de marketing dirigidas.

**Análisis de Datos:** La recopilación de datos detallados sobre usuarios y sus preferencias de viaje permite realizar análisis para mejorar los servicios y ofertas.

En resumen, la base de datos de Full Day Scape es una herramienta valiosa que contribuye significativamente a la optimización de los procesos operativos y de marketing de la agencia de viajes, garantizando la satisfacción del cliente y el éxito empresarial.

## **"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:

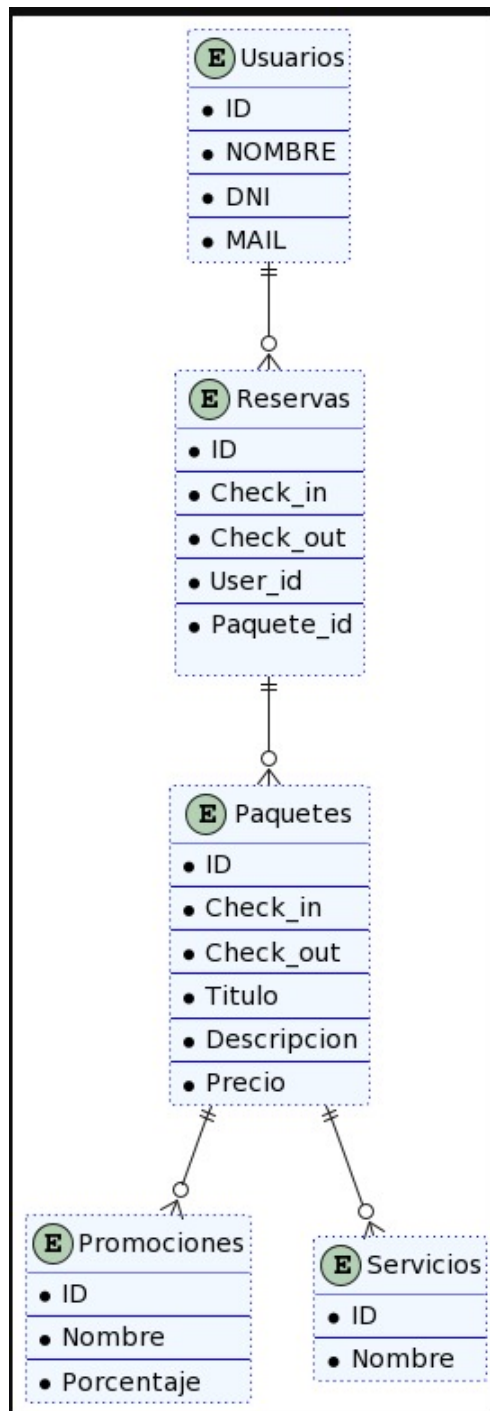
TABLA	NOMBRE ABREVIADO	NOMBRE DEL CAMPO COMPLETO	CLAVE PRIMARIA (PK)	CLAVE FORANEA (FK)	TIPO DE DATOS
USUARIOS	ID	IDENTIFICADOR DEL USUARIO	PK		INT
	NOMBRE	NOMBRE DEL USUARIO			VARCHAR
	MAIL	MAIL DEL USUARIO			VARCHAR
	DNI	DNI DEL USUARIO			INT
RESERVAS	ID	IDENTIFICADOR DE LA RESERVA	PK		INT
	PRECIO	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
PROMOCIONES	ID	IDENTIFICADOR DE LA PROMOCION	PK		INT
	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_PAQU ETES	IDENTIFICADOR DE LOG DE PAQUETES	PK		INT
	USUARIO_REAL IZADOR	USUARIO LOGUEADO EN LA BASE			VARCHAR
LOG_PAQUETES	TIPO_OPERACI ON	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_REALIZADOR	USUARIO LOGUEADO EN LA BASE			VARCHAR
LOG_RESERVAS	TIPO_OPERACION	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.

# **SENTENCIAS DCL (DATA CONTROL LANGUAGE)**

Las sentencias DCL (Data Control Language) se utilizan en bases de datos para gestionar aspectos de seguridad, como la creación de usuarios y la asignación o revocación de permisos. Los siguientes scripts son ejemplos de cómo se utilizan estas sentencias.

- **Script 1: CREATE\_USER\_VERGARA.sql**

Objetivo: Este script se enfoca en la creación de dos usuarios con diferentes propósitos en la base de datos.

## **Detalles:**

### **Creación del Usuario de Lectura:**

```
CREATE USER 'usuario_lectura'@'localhost' IDENTIFIED BY '123456';
```

Se crea un usuario llamado usuario\_lectura con contraseña '123456'. Este usuario está destinado a tener permisos solo de lectura.

### **Creación del Usuario de Escritura:**

```
CREATE USER 'usuario_escritura'@'localhost' IDENTIFIED BY '654321';
```

Se crea un segundo usuario, usuario\_escritura, con contraseña '654321'. Este usuario está pensado para tener permisos de lectura, escritura y posiblemente otros permisos adicionales.

- **Script 2: GRANT\_USUARIOS\_VERGARA.sql**

Objetivo: Este script asigna permisos específicos a los usuarios creados anteriormente.

## **Detalles:**

### **Permisos de Solo Lectura:**

```
GRANT SELECT ON fullday_scape.* TO 'usuario_lectura'@'localhost';
```

Se otorgan permisos de solo lectura (SELECT) al usuario usuario\_lectura para todas las tablas en la base de datos fullday\_scape.

### **Permisos de Lectura, Escritura y Modificación:**

```
GRANT SELECT, INSERT, UPDATE ON fullday_scape.* TO  
'usuario_escritura'@'localhost';
```

Al usuario usuario\_escritura se le otorgan permisos más amplios, incluyendo lectura (SELECT), inserción (INSERT) y actualización (UPDATE) en la misma base de datos.

### **Aplicación de los Permisos:**

```
FLUSH PRIVILEGES;
```

Este comando asegura que los cambios de permisos sean efectivos inmediatamente.

- **Script 3: REVOKE\_VERGARA.sql**

Objetivo: Este script se utiliza para revocar permisos específicos de los usuarios.

### **Detalles:**

#### **Revocación de Permisos de Eliminación:**

```
REVOKE DELETE ON fullday_scape.* FROM 'usuario_lectura'@'localhost';  
REVOKE DELETE ON fullday_scape.* FROM 'usuario_escritura'@'localhost';
```

A ambos usuarios, usuario\_lectura y usuario\_escritura, se les revoca el permiso de eliminar (DELETE) datos en todas las tablas de la base de datos fullday\_scape. Esto se hace para restringir la capacidad de eliminar registros, lo cual es crítico en muchas aplicaciones para mantener la integridad de los datos.

Estos scripts son fundamentales en la administración de bases de datos para garantizar que los usuarios tengan los permisos correctos según sus roles y responsabilidades.

## **SENTENCIAS TCL**

- **Script 1:** Gestión de Datos de Paquetes en Base de Datos

Objetivo: Este script está diseñado para manejar la eliminación de un registro específico (con ID 2) en una base de datos relacionada con paquetes turísticos, asegurando que todas las referencias a este registro en otras tablas sean gestionadas adecuadamente antes de su eliminación.

**Pasos del Script:**

Inicio de Transacción: Se inicia una transacción para asegurar que todas las operaciones se ejecuten como una unidad atómica.

**Actualización de Referencias:**

Se actualizan varias tablas (log\_paquetes, paquetes\_promociones, paquetes\_servicios) para establecer a NULL las referencias al paquete con ID 2. Esto es esencial para mantener la integridad referencial.

**Eliminación del Registro:**

Se elimina el registro con ID 2 de la tabla paquetes.

**Gestión de Transacción:**

La transacción puede ser confirmada (COMMIT) o revertida (ROLLBACK). En el script, estas líneas están comentadas para evitar su ejecución accidental.

Reinserción Opcional:

Se proporciona un script adicional comentado para insertar nuevamente el registro eliminado, si se requiere en el futuro.

- **Script 2:** Agregando Servicios y Manejo de SAVEPOINT

Objetivo: Este script se enfoca en agregar nuevos registros a la tabla servicios de una base de datos, utilizando la funcionalidad de SAVEPOINT para gestionar puntos específicos dentro de la transacción.

**Pasos del Script:**

Inicio de Transacción: Comienza una nueva transacción para agrupar las operaciones siguientes.

### **Inserción de Registros:**

Se insertan nuevos registros en la tabla servicios. El uso de INSERT IGNORE previene la inserción de duplicados.

Se agregan servicios como 'pileta', 'peluqueria', etc., con identificadores únicos.

### **Uso de SAVEPOINT:**

Se crea un SAVEPOINT (mi\_savepoint1) después de agregar los primeros cuatro registros. Esto permite un punto de restauración dentro de la transacción.

Se continúa insertando más registros.

### **Eliminación de SAVEPOINT:**

Se libera el SAVEPOINT creado, lo que implica que no se puede volver a este punto específico después de este paso.

### **Confirmación de la Transacción:**

Al igual que en el primer script, la línea para confirmar la transacción (COMMIT) está comentada para controlar su ejecución.

# **BACKUP DE BASE DE DATOS**

El script proporcionado es un ejemplo de un archivo de respaldo (backup) generado por MySQL para la base de datos fullday\_scape. Este tipo de archivo se utiliza para guardar una copia de seguridad de los datos contenidos en una base de datos, lo cual es crucial para la recuperación de datos en caso de pérdida o daño.

## **Uso del Script de Backup:**

### **Restauración de Datos:**

Este script puede ser utilizado para restaurar los datos de la base de datos fullday\_scape a su estado en el momento en que se realizó el respaldo.

### **Portabilidad:**

El archivo de respaldo puede ser utilizado para transferir datos entre diferentes servidores o entornos de MySQL.

## **CONCLUSIÓN**

La base de datos de Full Day Scape juega un papel crucial en el éxito operativo y comercial de la agencia de viajes, facilitando la gestión eficiente de clientes, paquetes, promociones y reservas. Al integrar detalladamente la información de los usuarios con una amplia gama de servicios y ofertas promocionales, esta base de datos no solo mejora la personalización y satisfacción del cliente, sino que también respalda estrategias de marketing efectivas y toma de decisiones basada en datos. En resumen, es una herramienta integral que impulsa la eficiencia, la adaptabilidad y el crecimiento en el ámbito competitivo del turismo.