

OBJETOS RENTACAR_BD

Vista **ActiveReservations**:

Esta vista, llamada **ActiveReservations**, tiene como objetivo listar todos los contratos de alquiler (**reservas**) que todavía no salieron, es decir, aquellos en los que la fecha de inicio (**start_date**) es mayor a la fecha actual (**CURDATE()**).

Cómo Funciona

Tablas Involucradas

La vista combina información de varias tablas mediante **joins**:

- **Contracts (c)**: Contiene los contratos de alquiler (datos principales como fechas y montos).
- **Clients (cl)**: Proporciona los datos del cliente (nombre y apellido).
- **Vehicles (v)**: Identifica el vehículo involucrado (por su matrícula o placa).
- **Agencies (a_pickup y a_return)**: Indica las agencias de recogida y devolución del vehículo.

Función **GetClientFullName**

Esta Función tiene como objetivo obtener el nombre completo de un cliente a partir de su primer y último nombre.

- La función toma el `client_id` como parámetro.
- Busca en la tabla **Clients** el primer nombre (`first_name`) y el apellido (`last_name`) del cliente con ese `client_id`.

Si quisieras mostrar el nombre completo de un cliente en una vista o procedimiento, podrías usar la función de esta forma

```
SELECT GetClientFullName(client_id) AS client_full_name FROM Clients;
```

Vista TodayContracts

Tiene como objetivo mostrar los contratos que inician en el día de hoy. Serviría para que cada sucursal sepa cuales son las salidas de día y organizarse.

- La vista selecciona todos los contratos de la tabla **Contracts** que tengan una fecha de inicio (start_date) igual a la fecha de hoy (CURDATE()).

Vista VehicleMaintenanceStatus

Tiene como objetivo mostrar el estado de mantenimiento de cada vehículo según su kilometraje.

- Se calcula el estado de mantenimiento basándose en el kilometraje actual del vehículo:
 - Si el kilometraje es $\geq 10,000$: Se considera "Mantenimiento Urgente".
 - Si el kilometraje es $\geq 8,000$: Se considera "Mantenimiento Próximo".
 - Si el kilometraje es $< 8,000$: Se considera "OK".
- Además, se incluye la última fecha de mantenimiento registrada para cada vehículo.

Ejemplo:

```
SELECT vehicle_id, status
FROM VehicleMaintenanceStatus
WHERE status = 'Mantenimiento Urgente';
```

Vista CustomerFeedbackFollowUp

Tiene como objetivo listar clientes con calificaciones bajas en las encuestas (≤ 3).

- Se seleccionan las encuestas donde la calificación del cliente es **menor o igual a 3**.
- Se ordenan los resultados por **fecha de la encuesta (survey_date)** de forma descendente para priorizar las más recientes.

Vista AgencyPerformance

Tiene como objetivo evaluar el desempeño de cada agencia.

- Se calcula la **cantidad total de contratos** gestionados por cada agencia.
- Se suma el **ingreso total generado** a partir de esos contratos.
- Se calcula el **promedio de ingresos por contrato**.
- Se obtiene la **calificación promedio** de las encuestas asociadas a cada agencia.

Función GetRemainingDays

Tiene como objetivo calcular cuántos días faltan para que termine un contrato.

- La función toma el contract_id como parámetro.
- Busca la fecha de finalización (end_date) del contrato en la tabla **Contracts**.
- Calcula la diferencia en días entre la fecha de finalización y la fecha actual (CURDATE()).

Ejemplo:

```
SELECT contract_id, GetRemainingDays(contract_id) AS remaining_days FROM Contracts;
```

Procedimiento GetIncomeByDateRange

Tiene como objetivo obtener los ingresos generados por los pagos dentro de un rango de fechas específico.

- El procedimiento toma dos parámetros de fecha: start_date y end_date.
- Primero, selecciona todos los pagos realizados en el rango de fechas proporcionado, mostrando detalles como el payment_id, el contrato asociado, el nombre del cliente, el monto pagado, y la agencia de recogida y devolución.
- Luego, calcula la **suma total de los ingresos** generados por esos pagos.

Ejemplo:

```
CALL GetIncomeByDateRange('2024-01-01', '2025-01-01');
```

Procedimiento: GetUpcomingContractReturns

Tiene como objetivo obtener los contratos que estén próximos a terminar en base al valor que le asignemos (cantidad de días a terminar)

- Este procedimiento tomará una fecha como parámetro (por ejemplo, CURDATE() + 7 para los próximos 7 días)
- Buscará todos los contratos que tienen una end_date en los próximos días.

Ejemplo:

```
CALL GetUpcomingContractReturns(7);
```

TRIGGER Contract_bd y TRIGGER Payments_bd

Estos triggers guardan la información de los contratos y pagos eliminados en una tabla de auditoría. Esto es útil para tener un historial de eliminaciones, en caso de que necesites restaurar la información o simplemente hacer un seguimiento.

Ejemplo:

```
DELETE FROM Payments
```

```
WHERE contract_id = 1;
```

```
DELETE FROM Contracts
```

```
WHERE contract_id = 1;
```

```
SELECT * FROM rentacar_db.deletedcontracts;
```

```
SELECT * FROM rentacar_db.deletedpayments;
```

TRIGGER InvalidContDates_au

El objetivo de este trigger es evitar que se inserten contratos con una fecha de inicio (start_date) posterior a la fecha de finalización (end_date). Si esto ocurre, el trigger genera un error y evita que se guarde el contrato.

Ejemplo:

```
INSERT INTO Contracts (contract_id, client_id, vehicle_id, start_date, end_date)  
VALUES (1, 1, 1, '2024-12-10', '2024-12-05');
```