TP SQL – Entrega 2

Objetos creados, propósito y pruebas

Base: agencias_growth_marketing • Para esta entrega, se continúa usando la BBDD usada para la entrega 1, con una pequeña modificación que se detalla a continuación. Al final del archivo, se expone el DER final.

Modificación respecto a la Entrega 1

En la primera entrega, la tabla *Cliente* contenía un campo `industria_ref` de tipo `VARCHAR` que almacenaba el nombre de la industria de manera textual. A partir de la corrección indicada por la profesora, se ajustó el diseño para cumplir con las buenas prácticas de normalización: ahora el campo se llama `industria_id` y es una clave foránea que referencia a `Industria(industria_id)`. De esta forma, se garantiza la integridad referencial y se evita la duplicación o inconsistencia de valores de texto. Los *INSERT* de clientes también fueron adaptados para utilizar subconsultas sobre la tabla *Industria*.

Orden de ejecución recomendado

1) Reset opcional: DROP DATABASE/CREATE DATABASE.

2) Ejecutar Trabajo Final Motto.sql. Para facilitar, se encuentra todo el desarrollo del trabajo en un solo código SQL. Este file incluye creación de BD, tablas, inserción de datos, creación de Vistas, funciones, stored procedures y triggers.

4) Correr los tests al final del script.

Resumen de objetos

Este documento describe las vistas, funciones, stored procedures y triggers implementados, incluyendo su objetivo, tablas involucradas y cómo verificarlos. En la siguiente tabla se describe cada uno de los objetos creados:

Tipo	Nombre	Propósito breve
Vista	v_agencia_resumen	Ficha con KPIs por agencia (servicios, industrias, certificaciones, equipo).
Vista	v_casos_detalle	Casos de éxito + KPI principal + conteo de métricas.

Vista	v_clientes_industria	Clientes con industria (legible) y cantidad de casos asociados.
Vista	v_agencia_capacidades	Mapa de capacidades: herramienta + tipo + nivel de uso por agencia.
Vista	v_agencia_rango_precios	Rango consolidado de precios por agencia y moneda de referencia.
Función	fn_cert_vigente	¿Agencia tiene certificación X vigente hoy? (BOOLEAN).
Función	fn_dias_contrato_restantes	Días restantes a fecha_fin (0 si venció; NULL si sin fin).
SP	sp_buscar_agencias	Buscador multifiltro (país, servicio, industria, tamaño equipo).
SP	sp_registrar_contrato	Alta de contrato y aceptación automática de propuesta (si aplica).
Trigger	trg_contrato_propuesta_aceptada	Al insertar contrato con propuesta, fuerza estado='Aceptada'.
Trigger	trg_validar_fechas_caso	Bloquea CasoExito con fecha_fin < fecha_inicio.

Vista: v_agencia_resumen

Propósito: KPIs resumidos por agencia para facilitar filtros y comparación.

Tablas involucradas: Agencia; subconsultas a AgenciaServicio, AgenciaIndustria, AgenciaCertificacion, MiembroEquipo.

Lógica clave: Cuenta servicios, industrias, certificaciones y miembros por agencia vía subconsultas correlacionadas. Incluye país, ciudad, tamaño de equipo y año de fundación.

Prueba rápida:

```
SELECT * FROM v_agencia_resumen LIMIT 5;
```

Vista: v_casos_detalle

Propósito: Detalle de casos con KPI principal y profundidad de medición.

Tablas involucradas: CasoExito JOIN Agencia; subconsulta a ResultadoCaso.

Lógica clave: Muestra título, periodo, KPI principal, valor, moneda, link de soporte y cantidad de métricas registradas en ResultadoCaso.

Prueba rápida:

```
SELECT * FROM v_casos_detalle ORDER BY fecha_inicio DESC LIMIT 5;
```

Vista: v_clientes_industria

Propósito: Clientes con industria legible y cantidad de casos asociados.

Tablas involucradas: Cliente LEFT JOIN Industria; subconsulta a CasoCliente.

Lógica clave: Mapea industria_id a nombre de Industria y contabiliza casos vinculados en la tabla puente CasoCliente.

Prueba rápida:

```
SELECT * FROM v_clientes_industria ORDER BY casos_asociados DESC, razon_social;
```

Vista: v_agencia_capacidades

Propósito: Mapa de capacidades tecnológicas por agencia.

Tablas involucradas: AgenciaHerramienta JOIN Agencia JOIN Herramienta.

Lógica clave: Expone herramienta, tipo (p. ej., 'CEP', 'Analytics', 'Ads') y nivel_uso declarado por agencia.

Prueba rápida:

```
SELECT * FROM v_agencia_capacidades WHERE herramienta LIKE 'Braze%' OR tipo='CEP';
```

Vista: v_agencia_rango_precios

Propósito: Rango consolidado de precios por agencia y moneda más frecuente.

Tablas involucradas: Agencia JOIN AgenciaPlan JOIN PlanPrecio.

Lógica clave: Calcula MIN(monto_min) y MAX(monto_max) por agencia. Selecciona moneda de referencia más frecuente entre sus planes.

Prueba rápida:

```
SELECT * FROM v_agencia_rango_precios ORDER BY monto_min_agencia;
```

Notas: Si existen múltiples monedas por agencia, se muestra la de mayor frecuencia como referencia.

Función: fn_cert_vigente(agencia_id INT, cert_nombre VARCHAR(150)) RETURNS BOOLEAN

Propósito: Indica si la agencia posee la certificación indicada vigente a la fecha actual.

Tablas involucradas: AgenciaCertificacion JOIN Certificacion.

Entradas (parámetros): agencia_id, cert_nombre

Salida: BOOLEAN (TRUE/FALSE)

Lógica clave: Verifica existencia de fila con nombre de certificación coincidente y rango de fechas que incluya la fecha actual. Tolera NULL en fecha_obtencion/fecha_expiracion como abiertos.

Prueba rápida:

```
SELECT fn_cert_vigente(1, 'Google Partner') AS vigente;

SELECT a.nombre FROM Agencia a WHERE fn_cert_vigente(a.agencia_id, 'Meta Partner');
```

Notas: Implementada como READS SQL DATA. Compatible con MySQL 8 (evita display width deprecado).

Función: fn_dias_contrato_restantes(contrato_id INT) RETURNS INT

Propósito: Devuelve días hasta fecha_fin (0 si venció; NULL si sin fecha de fin).

Tablas involucradas: Contrato.

Entradas (parámetros): contrato_id

Salida: INT (NULL posible)

Lógica clave: Lee fecha_fin de Contrato. Si es NULL → NULL; si ya venció → 0; si no, DATEDIFF(fecha_fin, CURRENT_DATE()).

Prueba rápida:

```
SELECT fn_dias_contrato_restantes(1) AS dias_restantes;

SELECT contrato_id, fn_dias_contrato_restantes(contrato_id) FROM Contrato LIMIT 10;
```

Stored Procedure: sp_buscar_agencias(p_pais, p_servicio_id, p_industria_id, p_min_equipo)

Propósito: Buscador con filtros opcionales; arma SQL dinámico y aplica solo los filtros provistos.

Tablas involucradas: Agencia (+ AgenciaServicio y/o AgenciaIndustria si se filtra por esas dimensiones).

Entradas (parámetros): p_pais VARCHAR(80) NULL, p_servicio_id INT NULL, p_industria_id INT NULL, p_min_equipo INT NULL

Salida: Resultset con agencia_id, nombre, país, ciudad, tam_equipo y contadores básicos.

Lógica clave: Construye SELECT base y agrega JOINs/WHERE según filtros no nulos. Usa PREPARE/EXECUTE y bind variable en combinaciones para evitar SQL injection.

Prueba rápida:

CALL sp_buscar_agencias('Argentina', NULL, NULL, 10);

CALL sp_buscar_agencias(NULL, 1, NULL, NULL); -- por servicio

CALL sp_buscar_agencias(NULL, NULL, 2, NULL); -- por industria

CALL sp_buscar_agencias(NULL, NULL, NULL, NULL); -- sin filtros

Notas: Agrupa por agencia para evitar duplicados cuando hay múltiples matches en tablas puente.

Stored Procedure: sp_registrar_contrato(p_agencia_id, p_cliente_id, p_propuesta_id, p_fecha_inicio, p_fecha_fin, p_monto_total, p_moneda, p_estado)

Propósito: Inserta contrato y, si referencia una propuesta, actualiza su estado a 'Aceptada'.

Tablas involucradas: Contrato; Propuesta.

Entradas (parámetros): IDs y datos del contrato; p_propuesta_id puede ser NULL; p_estado por defecto se usa 'Activo' si viene NULL.

Salida: Devuelve contrato_id_creado (SELECT final).

Lógica clave: INSERT en Contrato; si p_propuesta_id no es NULL → UPDATE Propuesta.estado='Aceptada'. Usa LAST_INSERT_ID() para exponer el ID creado.

Prueba rápida:

CALL sp_registrar_contrato(1, 1, 1, '2025-01-01', NULL, 5000.00, 'USD', 'Activo');

SELECT * FROM Propuesta WHERE propuesta_id = 1; -- verificar estado 'Aceptada'

Trigger: trg_contrato_propuesta_aceptada (AFTER INSERT ON Contrato)

Propósito: Al crear un contrato que referencia una propuesta, fuerza su estado a 'Aceptada'.

Tablas involucradas: Contrato; Propuesta.

Lógica clave: En AFTER INSERT, si NEW.propuesta_id no es NULL → UPDATE Propuesta SET estado='Aceptada' WHERE propuesta_id=NEW.propuesta_id.

Prueba rápida:

INSERT INTO Contrato(agencia_id, cliente_id, propuesta_id, fecha_inicio, monto_total, moneda, estado)

VALUES (1,1,2,'2025-01-10',3000,'USD','Activo');

SELECT estado FROM Propuesta WHERE propuesta_id=2; -- debe ser 'Aceptada'

Notas: Complementa la lógica del SP, garantizando consistencia aunque se cree el contrato por otras vías.

Trigger: trg_validar_fechas_caso (BEFORE INSERT ON CasoExito)

Propósito: Evita incoherencias temporales: fecha_fin no puede ser menor que fecha_inicio.

Tablas involucradas: CasoExito.

Lógica clave: En BEFORE INSERT, si NEW.fecha_fin < NEW.fecha_inicio → SIGNAL SQLSTATE '45000' con mensaje descriptivo.

Prueba rápida:

INSERT INTO CasoExito(agencia_id, titulo, descripcion, fecha_inicio, fecha_fin, kpi_principal, kpi_valor, moneda)

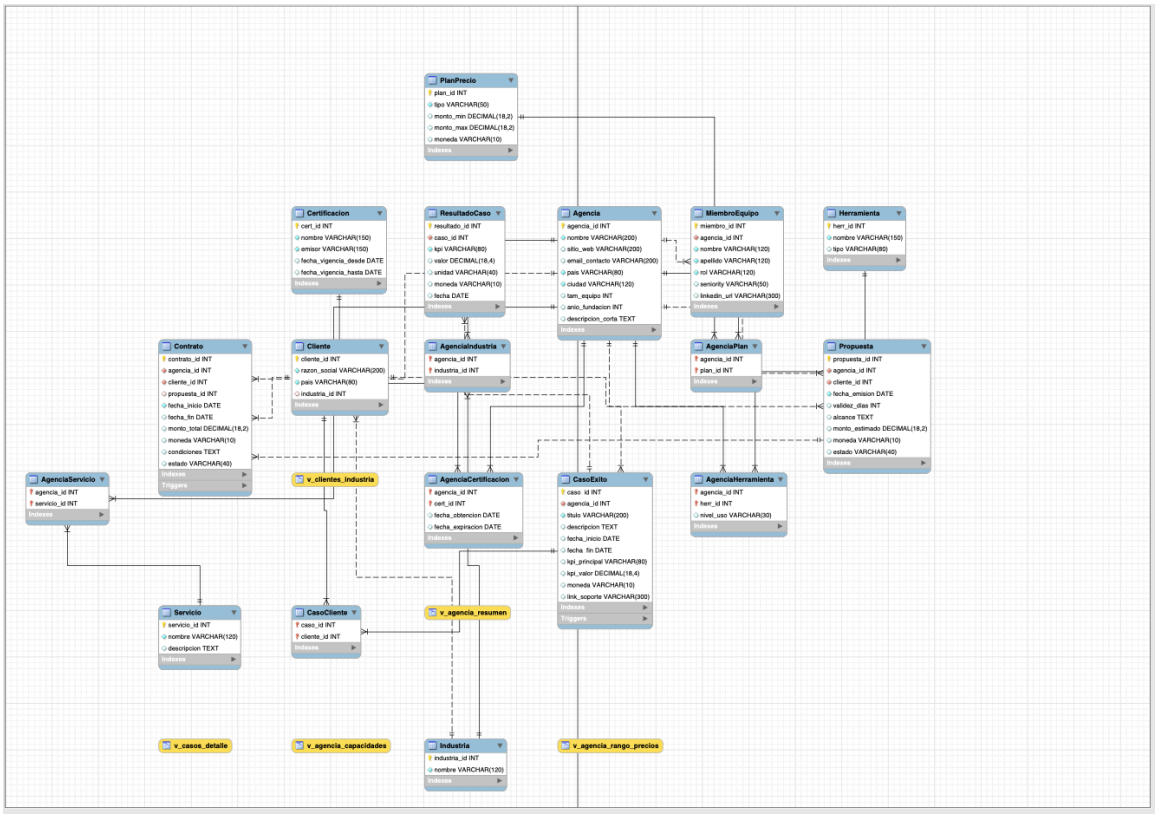
VALUES (1,'Test Fechas','Debe fallar','2025-02-01','2025-01-01','ROAS',2.0,'USD'); -- debe fallar

Notas: Se puede replicar lógica en BEFORE UPDATE si se permiten cambios de fechas.

Observación general

• Las definiciones incluyen bloques de limpieza (DROP IF EXISTS) para permitir re-ejecución segura.

Diagrama entidad - relación (DER)


PlanPrecio
plan_id INT
tipo VARCHAR(50)
monto_min DECIMAL(18,2)
monto_max DECIMAL(18,2)
moneda VARCHAR(10)
Indexes
Certificacion
cert_id INT
nombre VARCHAR(150)
emisor VARCHAR(150)
fecha_vigencia_desde DATE
fecha_vigencia_hasta DATE
Indexes
ResultadoCaso
resultado_id INT
caso_id INT
kpi VARCHAR(80)
valor DECIMAL(18,4)
unidad VARCHAR(40)
moneda VARCHAR(10)
fecha DATE
Indexes
Agencia
agencia_id INT
nombre VARCHAR(200)
sitio_web VARCHAR(200)
email_contacto VARCHAR(200)
pais VARCHAR(80)
ciudad VARCHAR(120)
tam_equipo INT
anio_fundacion INT
descripcion_corta TEXT
Indexes
MiembroEquipo
miembro_id INT
agencia_id INT
nombre VARCHAR(120)
apellido VARCHAR(120)
rol VARCHAR(120)
seniority VARCHAR(50)
linkedin_url VARCHAR(300)
Indexes
Herramienta
herr_id INT
nombre VARCHAR(150)
tipo VARCHAR(80)
Indexes
Contrato
contrato_id INT
agencia_id INT
cliente_id INT
propuesta_id INT
fecha_inicio DATE
fecha_fin DATE
monto_total DECIMAL(18,2)
moneda VARCHAR(10)
condiciones TEXT
estado VARCHAR(40)
Indexes
Triggers
Cliente
cliente_id INT
razon_social VARCHAR(200)
pais VARCHAR(80)
industria_id INT
Indexes
AgenciaIndustria
agencia_id INT
industria_id INT
Indexes
AgenciaPlan
agencia_id INT
plan_id INT
Indexes
Propuesta
propuesta_id INT
agencia_id INT
cliente_id INT
fecha_emision DATE
validez_dias INT
alcance TEXT
monto_estimado DECIMAL(18,2)
moneda VARCHAR(10)
estado VARCHAR(40)
Indexes
AgenciaServicio
agencia_id INT
servicio_id INT
Indexes
v_clientes_industria
AgenciaCertificacion
agencia_id INT
cert_id INT
fecha_obtencion DATE
fecha_expiracion DATE
Indexes
CasoExito
caso_id INT
agencia_id INT
titulo VARCHAR(200)
descripcion TEXT
fecha_inicio DATE
fecha_fin DATE
kpi_principal VARCHAR(80)
kpi_valor DECIMAL(18,4)
moneda VARCHAR(10)
link_soporte VARCHAR(300)
Indexes
Triggers
AgenciaHerramienta
agencia_id INT
herr_id INT
nivel_uso VARCHAR(30)
Indexes
Servicio
servicio_id INT
nombre VARCHAR(120)
descripcion TEXT
Indexes
CasoCliente
caso_id INT
cliente_id INT
Indexes
v_agencia_resumen
v_casos_detalle
v_agencia_capacidades
Industria
industria_id INT
nombre VARCHAR(120)
Indexes
v_agencia_rango_precios