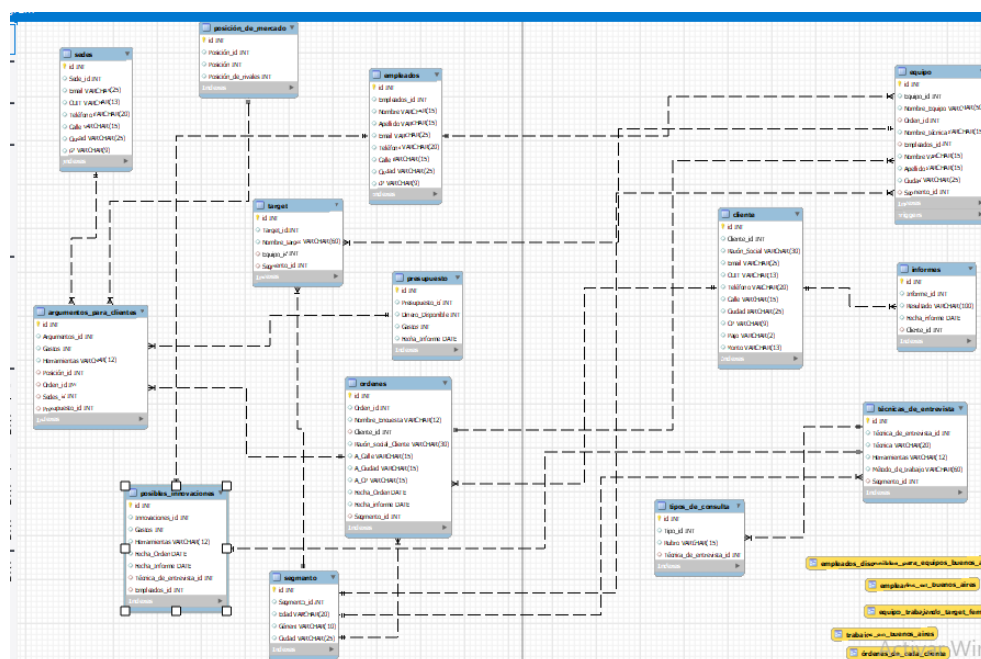


Proyecto de consultora privada de opinión pública y estudio de mercado

Hola, buenos días/tardes/noches, mi proyecto final es sobre el posible armado de una consultora privada que analice la opinión pública y las condiciones de mercado para poder cumplir mi sueño de trabajar en estudios sociológicos de dicho mercado, a la vez que pueda construir bases de datos para estudios futuros y principalmente actuales, respecto de conocer la opinión pública por un lado y por otro entender de qué manera se puede abordar una campaña de marketing efectiva, previamente a realizarla.

Es por ello que la armaría la base de datos utilizando los tipos de data como INT, VARCHAR, DATE, etc. para organizar la información que se pueda obtener sobre los segmentos trabajados para cumplir con las peticiones de los clientes y elaborar los informes correspondientes, esto también incluye, los datos sobre el cliente que pide la orden, datos de los empleados a los que se le otorga la orden que pasarán a conformar un equipo de investigación que son los que realizan la tarea solicitada en la orden y como ellos trabajaron con el target requerido para cumplir con dicha orden.

Respecto al diagrama de entidad relación que construí para cumplir con esto, dejare la siguiente imagen a continuación:



Si bien no se podrá ver claramente el diagrama debido al gran tamaño de este, posteriormente dejare un script detallado de cada tabla y sus conexiones para poder visualizarlo mediante ingeniería inversa y también un cuadro que detalla bien cada una de las tablas de la base de datos que cree para este proyecto final.

El objetivo de construir esta base de datos y trabajar de esta manera es el de poder, mediante la información construida a partir de los datos aportados por un target segmentado el cual debe estar definido previamente por el equipo, es estudiar ese segmento mediante distintas técnicas de entrevistas, almacenar esa información y realizar un informe para que el cliente pueda utilizar y así la consultora vender esa base de datos que el cliente ordenó y la cual servirá para conocer las condiciones del mercado, que lo podrían ayudar a optimizar su estrategias de mercantilización y al mismo tiempo a la consultora a ser más atractiva de atraer clientes siendo la experiencia y la calidad del equipo los principales argumentos para convencer nuevos clientes, los cuales pagarían los servicios de la empresa para obtener sus propias bases de datos y con un presupuesto creciente ir desarrollando innovaciones o construcción de sedes, que potencien la posición de mercado de la consultora.

Tabla	Campo Abreviado	Nombre del Campo completo	Clave Primaria (PK)	Clave Foranea (FK)	Tipo de Datos
Cliente	id_Cliente	Identificador del cliente	PK		INT
	Razón_Social	Razón Social del cliente			VARCHAR (30)
	Email	Email del cliente			VARCHAR (25)
	CUIT	CUIT del cliente			VARCHAR (13)
	Teléfono	Telefono del cliente			VARCHAR (20)
	Calle	Calle del cliente			VARCHAR (15)
	Ciudad	Ciudad del cliente			VARCHAR (15)
	CP	Codigo postal del cliente			VARCHAR (9)
	Pagó	Si el cliente realizo el pago correspondiente			VARCHAR (2)
	Monto	El monto a cobrar			INT

Segmento	Id_Segmento	Identificador del segmento trabajado	PK		INT
	Edad	Edades del segmento trabajado			VARCHAR (20)
	Género	Género del segmento trabajado			VARCHAR (10)
	Ciudad	Ciudades del segmento trabajado			VARCHAR (25)
	id_Orden	Identificador de la Orden	PK		INT
	id_Segmento	Identificador del Segmento			INT
	Nombre_Encuesta	Tipo de encuesta a realizar			VARCHAR (12)
	id_Cliente	Identificador del Cliente		FK	INT
	Razón Social_Cliente	Razón Social del cliente			VARCHAR (30)
Órdenes	A_Calle	Calle de envío del informe			VARCHAR (15)
	A_Ciudad	Ciudad de envío del informe			VARCHAR (10)
	A_CP	Codigo postal de envío del pedido			VARCHAR (9)
	Fecha_Orden	Fecha en que se realizo la orden			DATE
	Fecha_Informe	Fecha en la que se debera enviar el informe			DATE
	id_Empleados	Identificador de los empleados	PK		INT
	Nombre	Nombre del empleado			VARCHAR (15)
	Apellido	Apellido del empleado			VARCHAR (15)
	Email	Email del empleado			VARCHAR (25)

Empleados	DNI	DNI del empleado			VARCHAR (8)
	Teléfono	Telefono del empleado			VARCHAR (20)
	Calle	Calle del empleado			VARCHAR (15)
	Ciudad	Ciudad del empleado			VARCHAR (15)
	CP	Codigo postal del empleado			VARCHAR (9)
	id_Equipo	Identificador del Equipo	PK		INT
	Nombre_Equipo	Nombre del Equipo			VARCHAR (50)
	id_Orden	Identificador de la Orden		FK	INT
Equipo	id_Segmento	Identificador del Segmento			INT
	Nombre_Técnica	Nombre de la tecnica de entrevista			VARCHAR (15)
	id_empleados	Identificador de los empleados		FK	INT
	Nombre	Nombre del empleado			VARCHAR (15)
	Apellido	Apellido del empleado			VARCHAR (15)
	Ciudad	Ciudad del emplaedo			VARCHAR (15)
	id_Target	Identificador del Target	PK		INT
Target	id_Segmento	Identificador del Segmento			INT
	Nombre_Target	Nombre del Target			VARCHAR (60)
	id_Equipo	Identificador del Equipo		FK	INT
	Técnicas_de_entrevista_id	Identificador del tipo de técnica de entrevista	PK		INT
	Técnica	Técnica a aplicar			VARCHAR (20)
Técnicas de entrevista	Herramientas	Herramientas a aplicar			VARCHAR (12)

	Método_de_trabajo	Métodos a desarrollar en el trabajo			VARCHAR (60)
	id_Segmento	Identificador del segmento		FK	INT
	id_Informe	Identificador del informe	PK		INT
Informes	Resultado	Resultado del informe			VARCHAR (100)
	Fecha_informe	Fecha para entregar el informe			DATE
	Id_Cliente	Identificador de cliente		FK	INT
	Id_Presupuesto	Identificador de presupuesto	PK		INT
Presupuesto	Dinero_Disponible	Dinero disponible en el presupuesto			INT
	Gastos	Gastos en el presupuesto			INT
	Fecha_Informe	Fecha para entregar el informe			DATE
Sedes	Id_Sede	Identificador de la Sede	PK		INT
	Email	Email de la sede			VARCHAR (25)
	CUIT	Apellido del empleado			VARCHAR (13)
	Teléfono	Telefono de la Teléfono			VARCHAR (20)
	Calle	Calle de la Sede			VARCHAR (15)
	Ciudad	Ciudad de la Sede			VARCHAR (25)
	CP	Codigo postal de la Sede			VARCHAR (9)
	Id_Posibles_innovaciones	Identificador de las posibles innovaciones	PK		INT
	Gastos	Gastos necesarias			INT
	Herramientas	Herramientas adquiridas			VARCHAR (12)

Posibles innovaciones	Fecha_orden	Fecha de que se emitió la orden			DATE
	Fecha_informe	Fecha para entregar el Informe			DATE
	Id_Técnica_de_entrevista	Identificador de la entrevista		FK	INT
	Id_empleados	Identificador de los empleados		FK	INT
Posición de Mercado	Id_Posición de mercado	Identificador de posición de mercado	PK		INT
	Posición	Posición de mercado			INT
	Posición_de_rivales	Posición de mercado de los rivales			INT
Argumentos para clientes	Id_Argumentos_para_clientes	Identificador de argumentos para los clientes	PK		INT
	Gastos	Gastos que se llevan a cabo			INT
	Herramientas	Herramientas disponibles			VARCHAR (12)
	Id_Posición	Identificador de posición de mercado		FK	INT
	Id_Orden	Identificador de orden		FK	INT
	Id_Sede	Identificador de sede		FK	INT
	Id_Presupuesto	Identificador de presupuesto		FK	INT
Tipos de consulta	Id_Tipos_de_consulta		PK		INT
	Rubro				VARCHAR (15)
	Id_Técnica_de_entrevista			FK	INT

En esta tabla vemos los distintos campos que tienen las tablas y cómo se relacionan las tablas como también bajo qué tipo de data ingresa cada una para poder realizar una correcta carga de datos.

Y para finalizar pasaré el script con el que construir la base de datos y los modelos de der:

```
CREATE TABLE cliente(  
    id            INT NOT NULL auto_increment,  
    cliente_id    INT,  
    razón_social  VARCHAR(30),  
    email         VARCHAR(25),  
    cuit          VARCHAR(13),  
    teléfono     VARCHAR(20),  
    calle         VARCHAR(15),  
    ciudad        VARCHAR(25),  
    cp           VARCHAR(9),  
    pago         VARCHAR(2),  
    monto        VARCHAR(13),  
    PRIMARY KEY (id) );  
  
-- Tabla de Segmentos trabajados  
CREATE TABLE segmento(  
    id            INT NOT NULL auto_increment,  
    segmento_id  INT,  
    edad         VARCHAR (20),  
    género       VARCHAR (10),  
    ciudad       VARCHAR (25),  
    PRIMARY KEY (id) );  
  
-- Tabla de las Ordenes activas  
CREATE TABLE ordenes(  
    id            INT NOT NULL auto_increment,  
    orden_id     INT,  
    nombre_encuesta VARCHAR(12),  
    cliente_id    INT,  
    razón_social_cliente VARCHAR(30),  
    a_calle       VARCHAR(15),  
    a_ciudad      VARCHAR (15),  
    a_cp          VARCHAR (15),  
    fecha_orden   DATE,  
    fecha_informe DATE,
```

```

    segmento_id          INT,
    FOREIGN KEY (segmento_id) REFERENCES segmento (id),
    FOREIGN KEY (cliente_id) REFERENCES cliente (id),
    PRIMARY KEY (id) );

```

-- Tabla de empleados

```

CREATE TABLE empleados(
    id          INT NOT NULL auto_increment,
    empleados_id INT,
    nombre      VARCHAR(15),
    apellido    VARCHAR(15),
    email       VARCHAR(25),
    teléfono    VARCHAR(20),
    calle       VARCHAR(15),
    ciudad      VARCHAR(25),
    cp          VARCHAR(9),
    PRIMARY KEY (id) );

```

-- Tabla del equipo

```

CREATE TABLE equipo(
    id          INT NOT NULL auto_increment,
    equipo_id   INT,
    nombre_equipo VARCHAR(50),
    orden_id    INT,
    nombre_técnica VARCHAR(15),
    empleados_id INT,
    nombre      VARCHAR(15),
    apellido    VARCHAR(15),
    ciudad      VARCHAR(25),
    segmento_id INT,
    FOREIGN KEY (segmento_id) REFERENCES segmento (id),
    FOREIGN KEY (orden_id) REFERENCES ordenes (id),
    FOREIGN KEY (empleados_id) REFERENCES empleados (id),
    PRIMARY KEY (id) );

```

-- Tabla del target idóneo para el cliente

```

CREATE TABLE target(

```



```

id          INT NOT NULL auto_increment,
target_id   INT,
nombre_target VARCHAR (60),
equipo_id   INT,
segmento_id INT,
FOREIGN KEY (segmento_id) REFERENCES segmento(id),
FOREIGN KEY (equipo_id) REFERENCES equipo(id) ON DELETE CASCADE,
PRIMARY KEY (id));

```

-- Tabla de las técnicas de entrevistas disponibles

```

CREATE TABLE técnicas_de_entrevista(
    id          INT NOT NULL auto_increment,
    técnica_de_entrevista_id INT,
    técnica      VARCHAR (20),
    herramientas VARCHAR (12),
    método_de_trabajo VARCHAR (60),
    segmento_id  INT,
    FOREIGN KEY (segmento_id) REFERENCES segmento(id),
    PRIMARY KEY (id) );

```

-- Tabla de informes a entregar

```

CREATE TABLE informes (
    id          INT NOT NULL auto_increment,
    informe_id   INT,
    resultado    VARCHAR (100),
    fecha_informe DATE,
    cliente_id   INT,
    FOREIGN KEY (cliente_id) REFERENCES cliente(id),
    PRIMARY KEY (id) );

```

-- Tabla de presupuesto

```

CREATE TABLE presupuesto(
    id          INT NOT NULL auto_increment,
    presupuesto_id INT,
    dinero_disponible INT,
    gastos       INT,
    fecha_informe DATE,

```

```
PRIMARY KEY (id) );
```

```
-- Tabla de sedes
```

```
CREATE TABLE sedes (  
    id          INT NOT NULL auto_increment,  
    sede_id     INT,  
    email       VARCHAR(25),  
    cuit        VARCHAR(13),  
    teléfono    VARCHAR(20),  
    calle       VARCHAR(15),  
    ciudad      VARCHAR(25),  
    cp          VARCHAR(9),  
    PRIMARY KEY (id) );
```

```
-- Tabla de posibles innovaciones que se podrían realizar
```

```
CREATE TABLE posibles_innovaciones (  
    id                      INT NOT NULL auto_increment,  
    innovaciones_id        INT,  
    gastos                  INT,  
    herramientas           VARCHAR (12),  
    fecha_orden             DATE,  
    fecha_informe           DATE,  
    técnica_de_entrevista_id INT,  
    empleados_id           INT,  
    FOREIGN KEY (técnica_de_entrevista_id) REFERENCES técnica_de_en  
trevista_id  
    (id),  
    FOREIGN KEY (empleados_id) REFERENCES empleados (id),  
    PRIMARY KEY (id) );
```

```
-- Tabla de la posición en el mercado
```

```
CREATE TABLE posición_de_mercado (  
    id                      INT NOT NULL auto_increment,  
    posición_id            INT,  
    posición               INT,  
    posición_de_rivales    INT,  
    PRIMARY KEY (id));
```

-- Tabla de argumentos para los clientes

```
CREATE TABLE argumentos_para_clientes(  
    id                INT NOT NULL auto_increment,  
    argumentos_id     INT,  
    gastos            INT,  
    herramientas      VARCHAR (12),  
    posición_id       INT,  
    orden_id          INT,  
    sedes_id          INT,  
    presupuesto_id    INT,  
    FOREIGN KEY (posición_id) REFERENCES posición_de_mercado (id),  
    FOREIGN KEY (orden_id) REFERENCES ordenes (id),  
    FOREIGN KEY (sedes_id) REFERENCES sedes (id),  
    FOREIGN KEY (presupuesto_id) REFERENCES presupuesto (id),  
    PRIMARY KEY (id));
```

-- Tabla de los tipos de consulta

```
CREATE TABLE tipos_de_consulta(  
    id                INT NOT NULL auto_increment,  
    tipo_id           INT,  
    rubro             VARCHAR (15),  
    técnica_de_entrevista_id INT,  
    FOREIGN KEY (técnica_de_entrevista_id) REFERENCES técnicas_de_e  
ntrevista  
    (id),  
    PRIMARY KEY (id)  
);
```

```
INSERT INTO cliente(id, cliente_id, razón_social, email, cuit,  
teléfono, calle, ciudad, cp) VALUES
```

```
(1, 1, 'Burger King', 'burgerking@hotmail', '24-40797152-4',  
'1568157317', 'Burgercalle', 'Buenos Aires', '1609'),  
(2, 2, 'Mostaza', 'mostaza@hotmail', '25-40797153-5', '1568157318',  
'Mostazacalle', 'Buenos Aires', '1619'),  
(3, 3, 'Subway', 'Subway@hotmail', '26-40797154-6', '1568157319',  
'Subcalle', 'Santa Fe', '1215');
```

```
INSERT INTO segmento(id, segmento_id, edad, género, ciudad)
VALUES
```

```
(1, 1, 'Entre 20 y 40 años', 'Varones', 'Ciudad de Buenos Aires'),
(2, 2, 'Entre 40 y 59 años', 'Mixto', 'Ciudad de Buenos Aires'),
(3, 3, 'Entre 20 y 40 años', 'Mujeres', 'Santa Fe');
```

```
INSERT INTO ordenes(id, segmento_id, nombre_encuesta, cliente_id,
razón_social_cliente, a_calle, a_ciudad, a_cp, fecha_orden,
fecha_informe) VALUES
```

```
(1, 1, 'KingBurger', '1', 'BurgerKing', 'king1234',
'Buenos Aires', '1609', '2022-12-01', '2023-06-01'),
(2, 3, 'Mostaza', '2', 'Mostaza', 'Mostaza1234', 'Buenos Aires',
'1619', '2022-12-02', '2023-06-02'),
(3, 2, 'Subs', '3', 'Subway', 'Sub1234', 'Santa Fe', '1215', '2022-
12-03', '2023-06-03');
```

```
INSERT INTO empleados (id, empleados_id, nombre, apellido, email,
teléfono, calle, ciudad, cp) VALUES
```

```
(1, 1, 'Iván', 'Rojt', 'ivanrojt@hotmail', '1568157317', 'Calle1',
'Buenos Aires', 'B1609'),
(2, 2, 'Federico', 'Rojt', 'federicorajt@hotmail', '1568157318',
'Calle2', 'Buenos Aires', 'B1610'),
(3, 3, 'Álvaro', 'Rodriguez', 'ÁlvaroRodriguez@hotmail',
'1568157339', 'Calle3', 'Buenos Aires', 'b1649');
```

```
INSERT INTO equipo (id, nombre_equipo, orden_id, segmento_id,
nombre_técnica, empleados_id, nombre, apellido, ciudad) VALUES
```

```
(1, 'Investigación comida rapida1', 1, 1, 'Encuesta', 1, 'Iván',
'Rojt', 'Buenos Aires'),
(2, 'Investigación comida rapida2', 2, 2, 'Encuesta', 2, 'Federico',
'Rojt', 'Buenos Aires'),
(3, 'Investigación comida rapida3', 3, 3, 'Entrevista', 3, 'Álvaro',
'Rodriguez', 'Santa Fe');
```

```
INSERT INTO target (id, segmento_id, nombre_target, equipo_id)
VALUES
```

```
(1, 1, 'Personas varones entre 20 y 40 años', 1),
(2, 3, 'Personas varones entre 40 y 59 años', 1),
(3, 2, 'Personas mujeres entre 20 y 40 años', 1);
```

```
INSERT INTO técnicas_de_entrevista(id, técnica_de_entrevista_id,
técnica, herramientas, método_de_trabajo, segmento_id) VALUES
```

```
(1, 1, 'Encuesta', 'Guía', 'Online', 1),
(2, 3, 'Entrevista', 'Teléfono', 'Telefónica', 2),
(3, 2, 'Entrevista', 'Guía', 'Presencial', 3);
```

```
INSERT INTO informes (id, informe_id, resultado, fecha_informe,
cliente_id) VALUES
```

```
(1, 1, '65% interesados en probar nuevos platos con otras carnes',
'2023-06-01', 1),
(2, 2, '80% interesados en extensión de platos infantiles', '2023-
06-02', 2),
(3, 3, '25% interesadas en platos con menos calorías y 60% interesadas
en extensión del menu', '2023-06-03', 3);
```

```
INSERT INTO presupuesto (id, dinero_disponible, gastos,
fecha_informe) VALUES
```

```
(1, '20000000', '-8000000', '2023-06-01'),
(2, '20000000', '-8000000', '2023-06-02'),
(3, '20000000', '-8000000', '2023-06-03');
```

```
INSERT INTO sedes (id, sede_id, email, cuit, teléfono, calle,
ciudad, cp) VALUES
```

```
(1, 1, 'SedeCABA@hotmail.com', '0033445566', '+541168157317',
'Calle1', 'Ciudad de Buenos Aires', 'C1004'),
(2, 2, 'SedeGBA@hotmail.com', '0033445566', '+541186513771',
'Calle2', 'San Isidro', 'B1609'),
```

```
(3, 3, 'SedeSantaFe@hotmail.com', '0033445566', '+34268157317',  
'Calle3', 'Santa Fe Capital', 'S3004');
```

```
INSERT INTO posibles_innovaciones (id, gastos, herramientas,  
fecha_orden, fecha_informe, técnica_de_entrevista_id, empleados_id)  
VALUES
```

```
(1, '-20000', 'Online', '2022-12-01', '2023-06-01', 1, 1),  
(2, '-30000', 'Prescencial', '2022-12-02', '2023-06-02', 2, 2),  
(3, '-30000', 'Telefónica', '2023-06-03', '2022-12-03', 3, 3);
```

```
INSERT INTO posición_de_mercado (id, posición_id, posición,  
posición_de_rivales) VALUES
```

```
(1, 1, 10, 1),  
(2, 2, 10, 7),  
(3, 3, 3, 1);
```

```
INSERT INTO argumentos_para_clientes (id, argumentos_id, gastos,  
herramientas, posición_id, orden_id, sedes_id, presupuesto_id)  
VALUES
```

```
(1, 1, '-20000', 'Online', 1, 1, 1, 1),  
(2, 2, '-30000', 'Prescencial', 2, 2, 2, 2),  
(3, 3, '-30000', 'Telefónica', 3, 3, 3, 3);
```

```
INSERT INTO tipos_de_consulta (id, tipo_id, rubro,  
técnica_de_entrevista_id) VALUES
```

```
(1, 1, 'Gastronomía', 1),  
(2, 2, 'Gastronomía', 2),  
(3, 3, 'Gastronomía', 3);
```

Como yo ya explique porque fue la elección de las tablas anteriormente, lo único que queda por agregar sería el tema de los INSERT INTO y de porque los elegí y que me aportan, yo elegí estos valores con la intención de dar forma y explicar cuál serían unos ejemplos posibles del modelo de negocio propuesto para mi consultora.

VISTAS:

-- Vista de trabajos que hay en Buenos Aires

```
CREATE VIEW trabajos_en_buenos_aires
AS
SELECT a_ciudad, a_cp, razón_social_cliente, fecha_orden
FROM ordenes c
INNER JOIN equipo o ON c.orden_id = o.orden_id
WHERE ciudad LIKE '%Buenos Aires%';
```

-- Vista de empleados que hay en Buenos Aires

```
CREATE VIEW empleados_en_buenos_aires
AS
SELECT c.nombre, c.apellido, c.email, c.teléfono, c.calle, c.ciudad,
c.cp
FROM empleados c
INNER JOIN equipo o ON c.empleados_id = o.empleados_id
WHERE o.ciudad LIKE '%Buenos Aires%';
```

-- Vista de los equipos trabajando con un target femenino

```
CREATE VIEW equipo_trabajando_target_femenino
AS
SELECT c.target_id, c.segmento_id, c.nombre_target, c.equipo_id
FROM target c
INNER JOIN equipo o ON c.equipo_id = o.equipo_id
WHERE nombre_target LIKE '%Personas mujeres%';
```

-- Vista de las órdenes de cada cliente

```
CREATE VIEW órdenes_de_cada_cliente
AS
SELECT c.cliente_id, c.razón_social, c.teléfono, c.ciudad, c.cp,
c.pago, c.monto
FROM cliente c
INNER JOIN ordenes o ON c.cliente_id = o.cliente_id
WHERE c.cliente_id = o.cliente_id;
```

-

- Vista de empleados disponibles para los equipos que trabajan en Buenos Aires

```

CREATE VIEW empleados_disponibles_para_equipos_buenos_aires
AS
SELECT c.equipo_id, c.nombre_equipo, c.nombre_técnica, c.ciudad
FROM equipo c
INNER JOIN empleados o ON c.empleados_id = o.empleados_id
WHERE c.ciudad LIKE '%Buenos Aires%';

```

Estas vistas, las elegí para ejemplificar como sería el método de buscar ciertas características, como pueden ser los trabajos que se están realizando actualmente en Buenos aires, qué equipos trabajan actualmente con cierto tipo de target, qué empleados están disponibles para incorporarse a ciertos equipos particulares y qué tipo de ordenes pidió cada cliente, el motivo de estas elecciones fueron para organizar el trabajo según la ciudad para hacer un seguimiento correcto, qué equipos están desarrollando la investigación en esa ciudad, qué empleados se podrían incorporar a ese equipo según si están en esa ciudad y ver el seguimiento de las órdenes activas de la empresa para tener presente los plazos de tiempo y si el cliente ha depositado el pago o no.

FUNCIONES Y FUNCIONES ALMACENADAS (O STORED PROCEDURES):

```

DELIMITER //

-- Función para designar empleados
CREATE FUNCTION designacion_de_empleados (ciudad_empleado varchar(15)
),ciudad_de_orden varchar(15))
returns varchar(2)
deterministic
BEGIN
DECLARE resultado VARCHAR(2);
SELECT * INTO resultado
FROM empleados
INNER JOIN ordenes ON id_empleado = o.id_empleado
WHERE a_ciudad = ciudad_empleado AND a_ciudad = ciudad_de_orden;
RETURN resultado;

END//

DELIMITER //

```



```

-- Función para asignar los targets
CREATE FUNCTION asignación_target (nombre_target varchar (60),
equipo_id int)
returns int
deterministic
BEGIN
DECLARE equipo INT;
SELECT * INTO nombre_target FROM target
INNER JOIN equipo ON equipo = equipo;
RETURN equipo;

END//

DELIMITER //

-- Función almacenada para calcular los ingresos del mes de Junio
CREATE PROCEDURE `sp_calcular_ingresos_junio` (out total integer)
BEGIN
DECLARE monto INT; DECLARE resultado FLOAT;
SELECT Count(*) INTO monto FROM cliente;

END//

DELIMITER //

-- Función para segmentar el target por género femenino
CREATE PROCEDURE `sp_target_femenino` (inout nombre_target varchar (
60))
BEGIN
DECLARE género FLOAT;
SELECT id INTO nombre_target
FROM target
WHERE nombre_target LIKE "personas mujeres";END//CREATE PROCEDURE `
sp_target_masculino` (inout nombre_target varchar (60))
BEGIN
DECLARE género FLOAT;
SELECT id INTO nombre_target
FROM target
WHERE nombre_target LIKE "personas varones";

```

END//

DELIMITER //

Acá se están realizando los primeros ejercicios de Funciones y Funciones almacenadas para corroborar que el script funcione correctamente y para que también en la eventualidad que se requiera actualizar, borrar, agregar o formular de otra forma la tabla, siempre este la posibilidad de hacerlo mediante el script.

Por un lado, las funciones que elegí fueron para ejemplificar como sería crear una función para designar empleados disponibles para las distintas ordenes según las ciudades donde se deben trabajar y así asignar a los empleados según la cercanía que tengan con esas ciudades determinado por su ciudad de residencia mientras que la otra función es para asignar los targets que deben trabajar cada equipo para llevar a cabo las ordenes de los clientes.

Mientras que por el otro lado las funciones almacenadas son para ejemplificar algunos ejercicios posibles que se puedan realizar en sql como ingresar los ingresos que tenga la consultora en un mes determinado para tener registros de los ingresos a lo largo del tiempo y la otra función es para ejemplificar como se puede construir la segmentación del target con los criterios definidos anteriormente.

TRIGGERS:

--

Trigger que actualiza la lista de empleados después de designarlos a un equipo

```
CREATE TRIGGER log_empleados_delete
```

```
BEFORE INSERT ON equipo
```

```
FOR EACH ROW
```

```
BEGIN
```

```
DELETE FROM empleados WHERE nombre = new.nombre;
```

```
DELETE FROM empleados WHERE apellido = new.apellido;
```

```
END;
```

```
//

DELIMITER ;

DELIMITER //

-- Trigger que actualiza el identificador de un empleado luego de
ser designado

CREATE TRIGGER log_empleados_update

AFTER UPDATE ON equipo

FOR EACH ROW

BEGIN

INSERT INTO empleados (empleados_id) VALUES (new.empleados_id);

end;

//
```

No mucho más que añadir que el hecho de que estos triggers son para automatizar ciertos procesos en el script para que, a la hora de modificar cosas en la base de datos, ésta vaya realizando mediante los triggers, funciones automáticas para facilitar el trabajo de las query que se realicen en un workbench, ya sea con ejemplos de actualizar la lista de empleados disponibles de forma automática para no tener que borrarlos de la lista de manera manual o que el identificador de este ya quede registrado en el sistema mediante el trigger para evitar confusiones, errores, etc.

TRANSACCIONES:

```
START TRANSACTION;

DELETE FROM equipo WHERE segmento_id = 1;

DELETE FROM equipo WHERE segmento_id = 2;

DELETE FROM equipo WHERE segmento_id = 3;

DELETE FROM equipo WHERE segmento_id = 4;
```

```

SET foreign_key_checks=0;
INSERT INTO equipo (equipo_id, nombre_equipo, segmento_id,
nombre_técnica, nombre, apellido, ciudad) VALUES (1,
'Investigación comida rapida1', 1, 'Encuesta', 'Iván', 'Rojt',
'Buenos Aires');
SET foreign_key_checks=1;

SET foreign_key_checks=0;
INSERT INTO equipo(equipo_id, nombre_equipo, segmento_id,
nombre_técnica, nombre, apellido, ciudad) VALUES (2,
'Investigación comida rapida2', 2, 'Encuesta', 'Federico', 'Rojt',
'Buenos Aires');
SET foreign_key_checks=1;

SET foreign_key_checks=0;
INSERT INTO equipo(equipo_id, nombre_equipo, segmento_id,
nombre_técnica, nombre, apellido, ciudad) VALUES (3,
'Investigación comida rapida3', 3, 'Entrevista', 'Álvaro',
'Rodriguez', 'Santa Fe');
SET foreign_key_checks=1;

SET foreign_key_checks=0;
INSERT INTO equipo(equipo_id, nombre_equipo, segmento_id,
nombre_técnica, nombre, apellido, ciudad)VALUES (4,
'Investigación comida rapida4', 4, 'Entrevista', 'Álvaro',
'González', 'Santa Fe');
SET foreign_key_checks=1;

ROLLBACK;

START TRANSACTION;

INSERT INTO cliente (pago, monto) VALUES ('si', 1500000);

INSERT INTO cliente (pago, monto) VALUES ('si', 2500000);

INSERT INTO cliente (pago, monto) VALUES ('si', 3000000);

INSERT INTO cliente (pago, monto) VALUES ('si', 4000000);

SAVEPOINT savepoint1;

```

```

INSERT INTO cliente (pago, monto) VALUES ('si', 5000000);

INSERT INTO cliente (pago, monto) VALUES ('si', 5000000);

INSERT INTO cliente (pago, monto) VALUES ('si', 3000000);

INSERT INTO cliente (pago, monto) VALUES ('si', 1000000);

SAVEPOINT savepoint2;

RELEASE savepoint savepoint1;

COMMIT;

```

Las transacciones obviamente cuentan con la finalidad de actualizar la base de datos de manera permanente mediante la escritura de líneas de código que ingrese el equipo con la finalidad de modificar la base de datos durante el transcurso del trabajo.

CREACIÓN DE USUARIOS:

```

DROP USER 'read_only@consultora';

flush privileges;

CREATE USER 'read_only@consultora' identified IDENTIFIED
BY 'password_read_only';

GRANT SELECT ON consultora.* TO 'read_only@consultora';

DROP USER 'read_insert_update@consultora';

flush privileges;

CREATE USER 'read_insert_update@consultora' identified BY 'password_
read_insert_update';

GRANT SELECT, INSERT, UPDATE
ON consultora.* TO 'read_insert_update@consultora';

```

Y en esta última sección es mostrar de qué manera se crean usuarios para los equipos de las distintas sedes para que ingresen los datos obtenidos del trabajo con los segmentos mediante distintas herramientas y técnicas de entrevistas y con los cuales se podrá hacer el informe.

Conclusiones:

Así que a forma de cierre la intención es mostrar una base de datos funcional aunque a muy pequeña escala para mostrar como ejemplo y mostrar de qué manera se podría montar distintas bases de datos en SQL para consultoras privadas con la intención de fomentar y facilitar el desarrollo y desempeño de estas y espero que este documento sea del agrado del lector y los convenza de la propuesta. En lo personal espero que el proyecto siga evolucionando en una primera instancia a una consultora privada aunque también trabaje en el sector público para asentar la firma, para ello sería necesario convencer a los primeros clientes mostrando la sólida estructura de la repuesta para financiar el inicio realizando unas primeras ordenes que le sean de utilidad. Posteriormente la idea ya pasaría a ir extendiendo los alcances de la consultora para poder elaborar informes de un mayor rango a las primeras ciudades donde se trabaje y mediante el uso de SQL poder automatizar actualizaciones de las bases de datos o modificar estas bases de datos a medida que la consultora se vaya expandiendo, de forma de que nunca sea necesario construir múltiples bases de datos sino solo ingresar los nuevos datos para reemplazar los viejos y que esta única base de datos se pueda observar de distintas maneras con las query para recortar a la información que sea solicitada en ese momento, las potencialidades de utilizar las query de base de datos para un rubro cuyas bases nacen de las bases de datos son demasiadas para enumerarlas y una propuesta de innovar la consultoras mediante el uso de sql podría ayudar a agilizar la carga de datos y reducir los márgenes de error que están aquejando a estas desde hace un tiempo, espero que la lectura haya sido de su agrado, saludos!.

Saludos, Iván

Links:

Link de Github: <https://github.com/IvanRojt97/Proyecto-final-consultora-Rojt>

Link de GoogleDrive:

<https://drive.google.com/drive/folders/1mp75Hqinuc2voRMCQbBMnattm5vsPsoo?hl=es>