

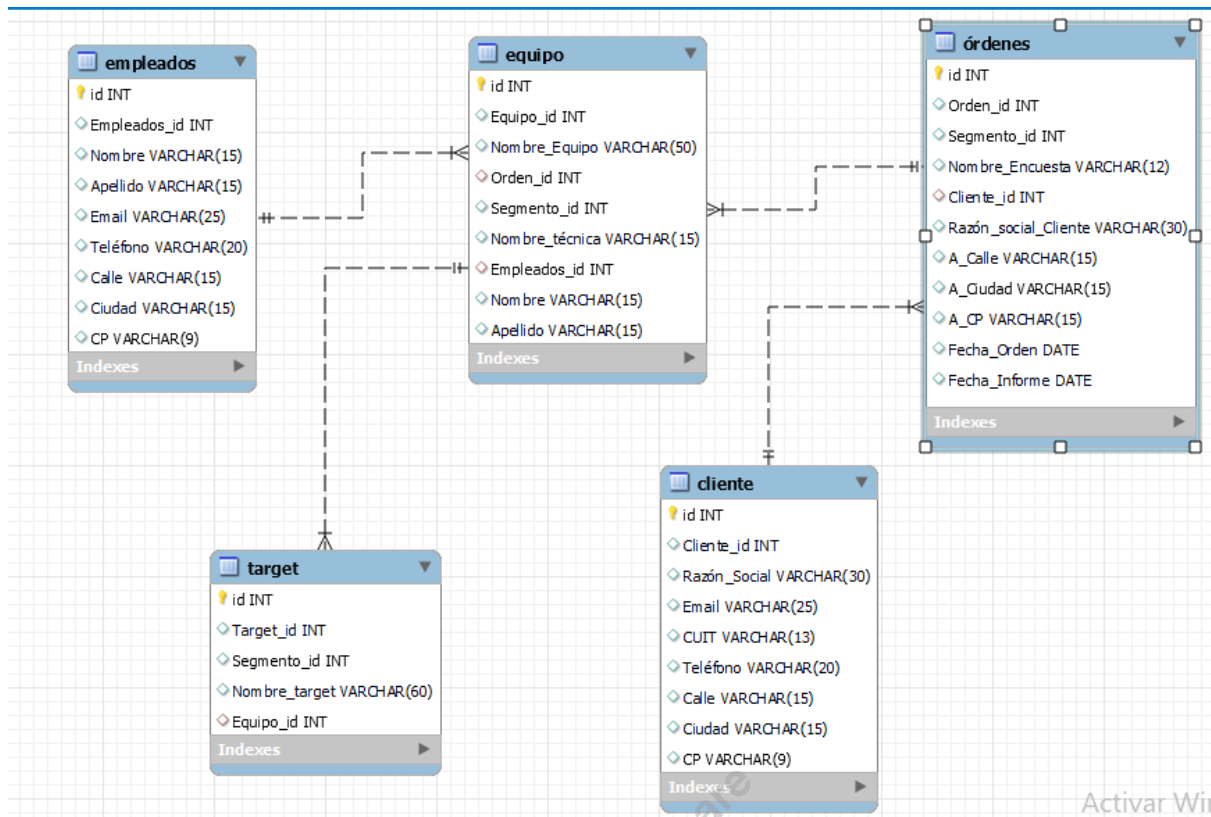
Proyecto de consultora privada de opinión pública y estudio de mercado

Mi arquetipo para el proyecto final en su etapa inicial es sobre el armado de una consultora privada que estudie la opinión pública y las condiciones de mercado para poder cumplir mi sueño de trabajar en los estudios sociológicos del mercado a la vez que puedo 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 previamente a realizarla.

Es por ello que la base de datos utilizando los tipos de data como int, varchar, date, etc. para organizar la información que se pueda obtener sobre, cuando la consultora tomó la orden de un cliente y cuando entregó el informe terminado, 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 investigación solicitada en la orden y como ellos trabajaron con el target requerido para cumplir con la orden.

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





El objetivo de realizar así el diagrama y trabajar de esta manera es el de poder, mediante la información construida a partir de los datos aportados por el target segmentado que se estudia, almacenar esa información y que el cliente la pueda utilizar y así la consultora vender esa base de datos que el cliente ordeno y con 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 que quieran pagar los servicios de la empresa para obtener sus propias bases de datos.

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
	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 envio del informe			VARCHAR (15)
	A_Ciudad	Ciudad de envio del informe			VARCHAR (10)
	A_CP	Codigo postal de envio 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			VARCHAR (8)

		empleado			
	Telefono	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 emplaeado			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

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 DATABASE consultora;

USE consultora;

TABLAS Y VALUES:

CREATE TABLE Cliente (

id INT PRIMARY KEY,

Cliente_id INT,

Razón_Social VARCHAR(30),

Email VARCHAR(25),

CUIT VARCHAR(13),

Teléfono VARCHAR(20),

Calle VARCHAR(15),

Ciudad VARCHAR(15),

CP VARCHAR(9),

Pagó VARCHAR(2),

Monto VARCHAR(13)

);

CREATE TABLE Órdenes (

id INT PRIMARY KEY,

Orden_id INT,

Segmento_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,  
  
FOREIGN KEY (Cliente_id) REFERENCES Cliente(id)  
  
);
```

```
CREATE TABLE Empleados (  
  
id INT PRIMARY KEY,  
  
Empleados_id INT,  
  
Nombre VARCHAR(15),  
  
Apellido VARCHAR(15),  
  
Email VARCHAR(25),  
  
Teléfono VARCHAR(20),  
  
Calle VARCHAR(15),  
  
Ciudad VARCHAR(15),  
  
CP VARCHAR(9)  
  
);
```



```
CREATE TABLE Equipo (  
  
id INT PRIMARY KEY,  
  
Equipo_id INT,  
  
Nombre_Equipo VARCHAR(50),  
  
Orden_id INT,  
  
Segmento_id INT,  
  
Nombre_técnica VARCHAR(15),  
  
Empleados_id INT,  
  
Nombre VARCHAR(15),  
  
Apellido VARCHAR(15),  
  
Ciudad VARCHAR(15),  
  
FOREIGN KEY (Orden_id) REFERENCES Órdenes(id),  
  
FOREIGN KEY (Empleados_id) REFERENCES Empleados(id)  
  
);
```

```
CREATE TABLE Target (  
  
id INT PRIMARY KEY,  
  
Target_id INT,  
  
Segmento_id INT,  
  
Nombre_target Varchar (60),
```

Equipo_id INT,

FOREIGN KEY (Equipo_id) REFERENCES Equipo(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 Órdenes (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','federicorojt@hotmail', '1568157318', 'Calle2', 'Buenos Aires', 'B1610'),

(3, 3, 'Álvaro','Rodriguez','ÁlvaroRogdriguez@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);

Como yo ya explique porque fue la elección de las tablas anteriormente, lo único que queda por agregar seria 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:

Create view Trabajos_Buenos_Aires as

Select A_Ciudad,A_CP,Razón_Social_Cliente,Fecha_Orden

From Órdenes c

INNER JOIN Equipo o ON c.Orden_id = o.Orden_id

Where Ciudad Like '%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%';

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%';

Create view órdenes_de_cada_cliente as

Select c.Cliente_id,c.Razón_Social,c.Teléfono,c.Ciudad,c.CP,c.Pagó,c.Monto

From Cliente c

INNER JOIN Órdenes o ON c.Cliente_id = o.Cliente_id

Where c.Cliente_id = o.Cliente_id;

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 con para ejemplificar como sería el tema de buscar ciertas características, como pueden ser los trabajos que se están realizando actualmente en Buenos aires, que equipos trabajan actualmente con cierto tipo de target, que empleados están disponibles para incorporarse a ciertos equipos particulares y que tipo de ordenes pidió cada cliente, el motivo de estas elecciones fueron para organizar el trabajo según la ciudad para hacer un seguimiento correcto, que equipos están desarrollando la investigación en esa ciudad, que empleados se podrían incorporar a ese equipo según si están en esa ciudad y ver el seguimiento de las ordenes 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 //
```

```
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 //
```

```
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 //
```

```
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 //
```

```
CREATE PROCEDURE `sp_calcular_ingresos_junio` (OUT total INTEGER)
```

```
BEGIN
```

```
DECLARE Monto INT;
```

```
DECLARE resultado FLOAT;
```

```
SELECT COUNT(*) INTO Monto FROM Cliente;
```

```
END //
```

```
DELIMITER //
```

```
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//
```

```
DELIMITER //
```

```
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//
```

Acá se están realizando los primero 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.

TRIGGERS:

```
CREATE TRIGGER Log_Pagó
```

```
AFTER INSERT ON órdenes
```

```
FOR EACH ROW
```

```
BEGIN
```

```
INSERT INTO cliente (pagó) Values (si);
```

```
END;
```

```
CREATE TRIGGER Log_Monto
```

```
BEFORE INSERT ON órdenes
```

```
FOR EACH ROW
```

```
BEGIN
```

```
INSERT INTO cliente (Monto) Values (NEW.monto (3000000));
```

```
END;
```

```
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;
```

```
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, esta vaya realizando mediante los triggers, funciones automáticas para facilitar el trabajo de las query que se realicen en un workbench.

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

Link de GoogleDrive:

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

Saludos, Ivan