



# PRUEBA SQL

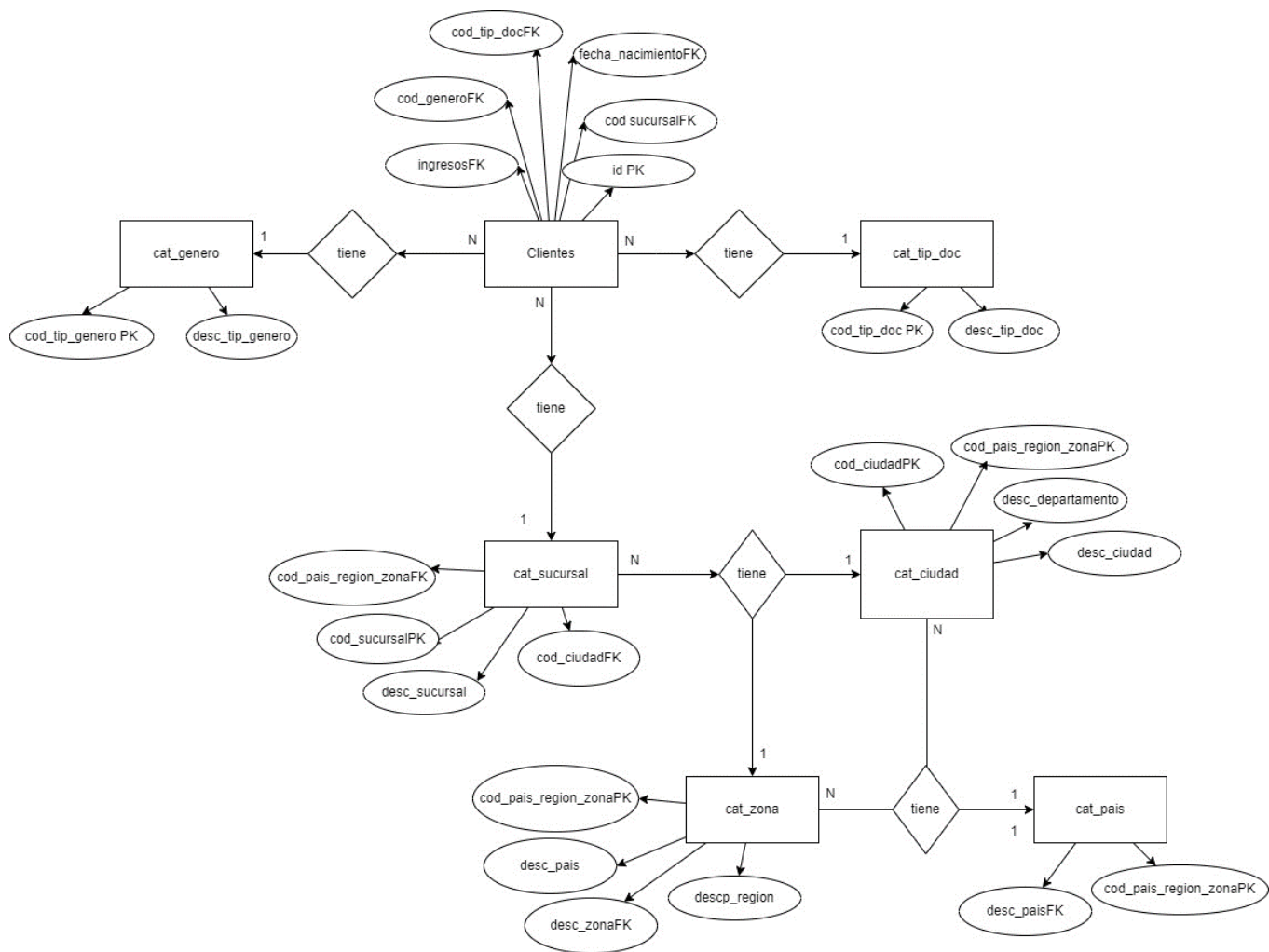
Darly Lorena Arias Santana

## **Descripción temática base de datos**

Esta base de datos tiene como enfoque información de unos clientes dirigida a una entidad financiera, ya que actualmente me desempeño como auxiliar administrativa en un banco, por lo que considero es un tema del cual tengo manejo y conocimiento y los análisis que adquiera con este trabajo podrían ser utilizados en un futuro en mi puesto, como una mejora.

Esta base contiene campos como ID, información de sucursales, ciudad, zona, país.

## Diagrama entidad de Relación



## Listado de tablas que componen la base de datos.

<b>Nombre tabla:</b> clientes						
contiene la información general de los clientes.						
CLAVE	NOMBRE CAMPO	TIPO	NOT NULL	UNIQUE	LONGITUD	DESCRIPCIÓN
PK	id	int	TRUE	TRUE	19	Identificación del cliente
FK	cod_tipo_doc	int	TRUE		1	Codigo de tipo documento
FK	cod_tip_genero	int	TRUE		1	codigo de genero del cliente
	fecha_nacimiento	date	TRUE		8	Fecha nacimiento del cliente
	ingresos	double	TRUE		13	monto ingresos del cliente
FK	cod_sucursal	int	TRUE		3	codigo de la sucursal a la que pertenece el cliente

<b>Nombre tabla:</b> cat_tip_doc						
contiene los codigos y descripción de tipos de documentos						
CLAVE	NOMBRE CAMPO	TIPO	NOT NULL	UNIQUE	LONGITUD	DESCRIPCIÓN
PK	cod_tip_doc	int	TRUE	TRUE	1	codigo tipo de documento
	desc_tip_doc	varchar	TRUE		30	descripción tipo de documentos

<b>Nombre tabla:</b> cat_genero						
contiene la información de los generos de los clientes						
CLAVE	NOMBRE CAMPO	TIPO	NOT NULL	UNIQUE	LONGITUD	DESCRIPCIÓN
PK	cod_tip_genero	int	TRUE	TRUE	1	codigo de genero
	desc_tip_genero	varchar	TRUE		30	descripción de genero

<b>Nombre tabla:</b> cat_sucursal						
contiene la información de la sucursal a la que pertenece cada cliente						
CLAVE	NOMBRE CAMPO	TIPO	NOT NULL	UNIQUE	LONGITUD	DESCRIPCIÓN
PK	cod_sucursal	int	TRUE	TRUE	3	codigo de sucursal
	desc_sucursal	varchar	TRUE		30	descripción de la sucursal
FK	cod_cuidad	int	TRUE		4	codigo de ciudad de la sucursal
FK	cod_pais_region_zona	int	TRUE		7	codigo de pais, región y zona de la sucursal

<b>Nombre tabla:</b> cat_ciudad						
contiene la información de la ciudad de la sucursal a la que pertenece cada cliente						
CLAVE	NOMBRE CAMPO	TIPO	NOT NULL	UNIQUE	LONGITUD	DESCRIPCIÓN
PK	cod_ciudad	int	TRUE	TRUE	4	codigo de ciudad
FK	desc_pais	varchar	TRUE		30	descripción del país
	desc_ciudad	varchar	TRUE		30	descripción de la ciudad
	desc_departamento	varchar	TRUE		30	descripción del departamento

<b>Nombre tabla:</b> cat_zona						
contiene la información de la zona de la sucursal a la que pertenece cada cliente						
CLAVE	NOMBRE CAMPO	TIPO	NOT NULL	UNIQUE	LONGITUD	DESCRIPCIÓN
PK	cod_pais_region_zona	int	TRUE	TRUE	7	codigo de ciudad
FK	desc_pais	varchar	TRUE		30	descripción del país
	desc_zona	varchar	TRUE		30	descripción de la zona
	desc_region	varchar	TRUE		30	descripción de la región

Nombre tabla:	cat_pais					
contiene la información del pais de la sucursal a la que pertenece cada cliente						
CLAVE	NOMBRE CAMPO	TIPO	NOT NULL	UNIQUE	LONGITUD	DESCRIPCIÓN
PK	cod_pais_region_zona	int	TRUE	TRUE	7	codigo del pais
FK	desc_pais	varchar	TRUE		30	descripción del país

## Script SQL

```
CREATE DATABASE `clientes`;
```

```
use clientes;
```

```
CREATE TABLE cat_tip_doc(
    cod_tip_doc int not null unique primary key,
    desc_tip_doc varchar(30) not null
);
```

```
CREATE TABLE cat_genero(
    cod_tip_genero int not null auto_increment primary key,
    desc_tip_genero varchar(30) not null
);
```

```
CREATE TABLE cat_pais(
    cod_pais_region_zona int not null unique primary key,
    desc_pais varchar(30) not null
);
```

```
CREATE TABLE cat_zona(
    cod_pais_region_zona int not null unique primary key,
    desc_pais varchar(30) not null,
    desc_zona varchar(30) not null,
    desc_region varchar(30) not null
);
```

);

```
CREATE TABLE cat_ciudad(  
    cod_ciudad int not null unique primary key,  
    cod_pais_region_zona int not null,  
    desc_ciudad varchar(30) not null,  
    desc_departamento varchar(30) not null  
);
```

```
CREATE TABLE cat_sucursal(  
    cod_sucursal int not null unique primary key,  
    desc_sucursal varchar(30) not null,  
    cod_ciudad int not null,  
    cod_pais_region_zona int not null  
);
```

```
CREATE TABLE clientes(  
    id int not null unique primary key,  
    cod_tip_doc int not null,  
    cod_tip_genero int not null,  
    fecha_nacimiento date not null,  
    ingresos double not null,  
    cod_sucursal int not null  
);
```

```
alter table clientes add foreign key (cod_tip_doc) references cat_tip_doc(cod_tip_doc);  
alter table clientes add foreign key (cod_tip_genero) references cat_genero(cod_tip_genero);  
alter table clientes add foreign key (cod_sucursal) references cat_sucursal(cod_sucursal);  
alter table cat_sucursal add foreign key (cod_ciudad) references cat_ciudad(cod_ciudad);
```

```
alter table cat_sucursal add foreign key (cod_pais_region_zona) references  
cat_zona(cod_pais_region_zona);
```

```
alter table cat_ciudad add foreign key (cod_pais_region_zona) references  
cat_pais(cod_pais_region_zona);
```

```
alter table cat_zona add foreign key (cod_pais_region_zona) references  
cat_pais(cod_pais_region_zona);
```

[Link github](#)

<https://github.com/darlyarias/darly-arias/blob/f62d31cb60481e9114e2d4642ed02dca947917a5/ideaArias.sql>

## Segunda entrega proyecto Final

### VISTAS:

1. vista\_ciudad: esta vista permite visualizar específicamente los clientes que registran bajo departamento del exterior “ Dpto del exterior”, esta vista está compuesta por la tabla cat\_ciudad, en la columna desc\_departamentos.

```
1  create view vista_ciudad as  
2  select desc_ciudad,desc_departamento from clientes.cat_ciudad  
3  where desc_departamento = 'Dpto del exterior';
```

2. Vista\_sucursal: Esta vista me permite identificar que clientes registran en las sucursales, específicamente en el “ cod\_sucursal” 1,2,3, esta vista esta compuesta por la tabla cat\_sucursal, usando la columna cod\_sucursal.

```
7  create view Vista_sucursal as  
8  select cod_sucursal,desc_sucursal from clientes.cat_sucursal  
9  where cod_sucursal in (1,2,3);
```

3. Vista\_documentos\_extranjeros: Esta vista me permite identificar los clientes con tipo de numero extranjeros, exactamente (2,5), Esta vista esta compuesta por la tabla cat\_tip\_doc usando la columna cod\_tip\_doc.

```
14 create view Vista_documentos_extranjeros as
15 select cod_tip_doc,desc_tip_doc from clientes.cat_tip_doc
16 where cod_tip_doc in (2,5);
--
```

4. Vista\_zonas\_colombia: Esta vista me permite identificar en la base de datos los clientes que pertenezcan al país "Colombia", esta vista esta compuesta por la tabla cat\_zona, usando la Columna desc\_pais.

```
21 create view Vista_zonas_colombia as
22 select desc_pais,desc_zona,desc_region from clientes.cat_zona
23 where desc_pais = "Colombia";
```

5. Vistas\_ingresos: esta vista me permite traer los datos de fecha de nacimiento y genero de los clientes que tengan ingresos iguales o superiores a "2000000", esta vista está compuesta por la tabla clientes, usando las columnas cod\_tip\_genero,fecha\_nacimiento,ingresos.

```
27 create view Vistas_ingresos as
28 Select cod_tip_genero,fecha_nacimiento,ingresos from clientes.clientes
29 where ingresos >= 2000000;
```



## FUNCIONES:

1. Función para contar las ciudades que pertenecen a un departamento, esta función específicamente me ayuda con la información de las ciudades pertenecientes a determinados departamentos y poder agruparlas según mi necesidad, esta función toma la tabla cat\_ciudad, usando las columnas desc\_departamento.

```
2 DELIMITER $$
3 #Se define como parametro de entrada el departamento al cual queremos contar las ciudades que se encuentran registradas en la BD
4 CREATE DEFINER=`root`@`localhost` FUNCTION `Contar_ciudad_por_departamento`(descripcion_departamento varchar(50)) RETURNS int
5 READS SQL DATA
6 BEGIN
7     DECLARE conteo INT;
8     #Construcción del query
9     SELECT
10         COUNT(*)
11     INTO conteo FROM
12         clientes.cat_ciudad
13     WHERE
14         desc_departamento = descripcion_departamento;
15
16     RETURN conteo;
17 END$$
18 DELIMITER ;
```

2. Función para obtener el nombre del país de la ciudad ingresada por medio de un join entre la tabla cat\_ciudad y cat\_pais, esta función específicamente me permite determinar el nombre de un país partiendo de la ciudad seleccionada, esta función toma la tabla cat\_ciudad, usando la columna desc\_ciudad y un join entre tabla cat\_ciudad y cat\_pais

```
23 DELIMITER $$
24 CREATE DEFINER=`root`@`localhost` FUNCTION `ObtenerPaisPorCiudad`(descripcion_ciudad VARCHAR(30)) RETURNS varchar(255) CHARSET utf8mb4
25 READS SQL DATA
26 BEGIN
27     DECLARE pais VARCHAR(255);
28
29     select p.desc_pais INTO pais
30     from clientes.cat_ciudad c
31     join clientes.cat_pais p ON c.cod_pais_region_zona = p.cod_pais_region_zona where c.desc_ciudad = descripcion_ciudad;
32     #el valor que permite crear la relación en ambas tablas es cod_pais_region_zona
33     RETURN pais;
34
35 END$$
36 DELIMITER ;
```

## STORE PROCEDURES

1. Creación procedimiento para ordenar una tabla, como parametros de entrada recibe el nombre de la tabla, el campo por el cual queremos ordenar y el metodo de ordenamiento: si es Ascendente se ingresa el valor ASC o si es descendente se ingresa el valor DESC.

```
1  #Creación procedimiento para ordenar una tabla
2  DELIMITER $$
3  CREATE PROCEDURE OrdenarTablas(
4  #como parametros de entrada recibe el nombre de la tabla, el campo por el cual queremos ordenar y el metodo de ordenamiento:
5      IN nombre_de_tabla VARCHAR(255),
6      IN nombre_de_campo VARCHAR(255),
7      IN orden VARCHAR(4)
8  )
9  BEGIN
10     SET @query = CONCAT('SELECT * FROM clientes.', nombre_de_tabla, ' ORDER BY ', nombre_de_campo, ' ', orden, ';');
11     PREPARE stmt FROM @query;
12     EXECUTE stmt;
13     DEALLOCATE PREPARE stmt;
14 END$$
15 DELIMITER ;
```

2. Creación procedimiento para eliminar un registro de la tabla sucursal, los parametros de entrada son el codigo de la sucursal.

```
18  #creación procedimiento para eliminar un registro de la tabla sucursal
19  DELIMITER $$
20  #los parametros de entrada son el codigo de la sucursal
21  CREATE DEFINER='root'@'localhost' PROCEDURE `eliminarSucursal`(
22      IN codigo_sucursal INT
23  )
24  BEGIN
25      SET @query = concat('delete from cat_sucursal where cod_sucursal = (',codigo_sucursal,')');
26      PREPARE stmt FROM @query;
27      EXECUTE stmt;
28      DEALLOCATE PREPARE stmt;
29  END$$
30  DELIMITER ;
```