



## Segunda Pre Entrega Proyecto final curso SQL Coderhouse

Ronnie Alvarez Castro

10/06/2024

### Base de datos para Salon de Belleza y Spa

Este proyecto es una base de datos para un curso de MySQL que permite gestionar información relacionada con un salon de belleza y spa, esta base de datos tiene relacion con productos, servicios, tratamientos, proveedores, facturas y clientes de una empresa MEL Coquette. La base de datos está diseñada para almacenar y procesar datos de manera eficiente, permitiendo realizar consultas y generar informes relevantes para la toma de decisiones.

**Estas son algunas de las tablas principales en esta base de datos:**

**clientes:** Almacena información de los clientes, como nombre, apellidos, email, teléfono, etc. Se relaciona con la tabla personas.

**empleados:** Guarda datos de los empleados, como tipo de contrato y salario. Se relaciona con la tabla personas.

**servicios:** Contiene información sobre los servicios ofrecidos en el salón.

**productos:** Almacena datos de los productos utilizados, como nombre, precio, cantidad, proveedor, etc. Se relaciona con la tabla unidades y proveedores.

**citas:** Guarda información sobre las citas de los clientes, como fecha, hora, empleado, servicio y detalles. Se relaciona con las tablas clientes, empleados y servicios.

**facturas:** Contiene datos sobre las facturas emitidas, como cliente, servicio, total, forma de pago, etc. Se relaciona con las tablas clientes, empleados y servicios.

**Proveedores:** Contiene información sobre los proveedores, como nombre, contacto, producto, teléfono. Se relaciona con las tablas de personas, productos y direcciones entre otras.

**Las relaciones entre las tablas permiten mantener la integridad y coherencia de los datos. Por ejemplo:**

Un cliente puede tener varias citas y facturas.

Un empleado puede atender varias citas.

Un servicio puede ser proporcionado a varios clientes.

Un producto puede ser provisto por varios proveedores.

En resumen, esta base de datos está diseñada para gestionar de manera eficiente la información relacionada con los clientes, empleados, servicios, productos, citas y facturas de un salón de belleza y spa, más adelante se detallan las tablas y sus relaciones así como las llaves primarias y foraneas.

[Link al GitHub de la Base de Datos](#)

[Link al Readme](#)

[Link al Modelo Entidad Relación](#)



## FUNCTIONS

### ■ Calcula el consumo por cliente

```
CREATE DEFINER='root'@'localhost' FUNCTION `calcular_consumo`(p_cliente_id INT) RETURNS decimal(10,2)
DETERMINISTIC
BEGIN
    DECLARE consumo DECIMAL(10,2);
    SELECT SUM(total_fact) INTO consumo
    FROM facturas
    WHERE cliente_id = p_cliente_id;
    set consumo=ifnull(consumo,0);
    RETURN consumo;
END
```

### ■ Calcula la edad del cliente

```
CREATE DEFINER='root'@'localhost' FUNCTION `calcular_edad`(fecha_nacimiento DATE) RETURNS int
DETERMINISTIC
BEGIN
    DECLARE edad INT;
    SET edad = YEAR(CURDATE()) - YEAR(fecha_nacimiento);
    IF (MONTH(CURDATE()) < MONTH(fecha_nacimiento))
        OR (MONTH(CURDATE()) = MONTH(fecha_nacimiento)
        AND DAY(CURDATE()) < DAY(fecha_nacimiento)) THEN
        SET edad = edad - 1;
    END IF;
    RETURN edad;
END
```

### ■ Calcula la utilidad por producto.

```
CREATE DEFINER='root'@'localhost' FUNCTION `calcular_ganancia`(p_producto_id INT) RETURNS decimal(10,2)
DETERMINISTIC
BEGIN
    DECLARE ganancia DECIMAL(10,2);
    if not exists(
        select 1 from productos where producto_id = p_producto_id
    ) then
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El producto no existe.';
    END IF;
    SELECT (precioVenta - precioCosto) INTO ganancia
    FROM productos
    WHERE producto_id = p_producto_id;
    RETURN ganancia;
END
```



■ **Calcula el total de ventas para un periodo dado hay que indicar la fecha inicial y la final**

```
CREATE DEFINER=`root`@`localhost` FUNCTION `total_ventas`(  
    fecha_inicial DATE,  
    fecha_final DATE  
) RETURNS decimal(10,2)  
    DETERMINISTIC  
BEGIN  
    DECLARE total_ventas DECIMAL(10,2);  
  
    SELECT FORMAT(SUM(`total_fact`), 2, 'es_CR') INTO total_ventas  
    FROM `facturas`  
    WHERE `fecha_pago` BETWEEN fecha_inicial AND fecha_final;  
  
    RETURN total_ventas;  
END
```

■ **Calcula las ventas por cliente**

```
CREATE DEFINER=`root`@`localhost` FUNCTION `ventas_x_cliente`(p_cliente_id INT) RETURNS decimal(10,2)  
    DETERMINISTIC  
BEGIN  
    DECLARE total DECIMAL(10,2);  
    if not exists(  
        select 1 from facturas where cliente_id = p_cliente_id  
    ) then  
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El cliente no ha consumido.';  
        set total =0.0;  
    END IF;  
    SELECT sum(total_fact) INTO total  
    FROM facturas  
    WHERE cliente_id = p_cliente_id;  
    RETURN total;  
END
```



## STORED PROCEDURES

- **Actualiza productos con la informacion nueva y verifica que existan los datos y sino envia mensaje de error**

```
CREATE DEFINER='root'@'localhost' PROCEDURE `actualizar_producto`(
    IN p_producto_id INT,
    IN p_nombre VARCHAR(100),
    IN p_precioCosto DECIMAL(10,2),
    IN p_precioVenta DECIMAL(10,2),
    IN p_cantidad DECIMAL(10,2),
    IN p_fechaCompra DATE,
    IN p_proveedor_id INT,
    IN p_unidad_id INT,
    IN p_productodetalle_id INT )
BEGIN
    IF NOT EXISTS (          -- verifica que el producto a modificar exista
        SELECT 1
        FROM productos
        WHERE producto_id = p_producto_id
    ) THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El producto no existe.';
    END IF;
    IF NOT EXISTS (          -- verifica que el proveedor digitado exista
        SELECT 1
        FROM proveedores
        WHERE proveedor_id = p_proveedor_id
    ) THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El proveedor no existe.';
    END IF;
    IF NOT EXISTS (          -- verifica la unidad de medida
        SELECT 1
        FROM unidades
        WHERE unidad_id = p_unidad_id
    ) THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La unidad de Medida no existe.';
    END IF;
    IF NOT EXISTS ( -- verifica si existe el detalle de producto
        SELECT 1
        FROM productodetalles
        WHERE productodetalle_id = p_productodetalle_id
    ) THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El detalle del producto no existe.';
    END IF;
    UPDATE productos
    SET nombre = p_nombre,
        precioCosto = p_precioCosto,
        precioVenta = p_precioVenta,
        cantidad = p_cantidad,
        fechaCompra = p_fechaCompra,
        proveedor_id = p_proveedor_id,
        unidadMedida_id = p_unidad_id,
        productodetalle_id = p_productodetalle_id
    WHERE producto_id = p_producto_id;
END
```



■ **Agrega productos nuevo**

```
CREATE DEFINER='root'@'localhost' PROCEDURE `agregar_producto`(
  IN p_nombre VARCHAR(100),
  IN p_precioCosto DECIMAL(10,2),
  IN p_precioVenta DECIMAL(10,2),
  IN p_cantidad DECIMAL(10,2),
  IN p_fechaCompra DATE,
  IN p_proveedor_id INT,
  IN p_unidadMedida_id INT,
  IN p_productodetalle TEXT
)
BEGIN
  DECLARE v_productodetalle_id INT;

  IF NOT EXISTS (
    SELECT 1
    FROM proveedores
    WHERE proveedor_id = p_proveedor_id
  ) THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El proveedor no existe.';
  END IF;
  IF NOT EXISTS (
    SELECT 1
    FROM unidades
    WHERE unidad_id = p_unidadMedida_id
  ) THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La unidad de medida no existe.';
  END IF;
  IF NOT exists (
    SELECT 1 FROM productodetalles where productodetalle= p_productodetalle)
  then
    -- Si el detalle no existe lo agrega a la tabla productodetalles y toma el
    -- ultimo id en la tabla como referencia
    INSERT INTO productodetalles (productodetalle) VALUES (p_productodetalle);
    SET v_productodetalle_id = LAST_INSERT_ID();
  end if;
  SET v_productodetalle_id = (SELECT productodetalle_id from productodetalles where productodetalle=
p_productodetalle);
  INSERT INTO productos (nombre, precioCosto, precioVenta, cantidad, fechaCompra, proveedor_id,
unidadMedida_id, productodetalle_id)
  VALUES (p_nombre, p_precioCosto, p_precioVenta, p_cantidad, p_fechaCompra, p_proveedor_id,
p_unidadMedida_id, v_productodetalle_id);
END
```



### ■ Elimina productos de la base de datos

```
CREATE DEFINER='root'@'localhost' PROCEDURE `eliminar_producto`(
  IN p_producto_id INT)
BEGIN
  IF NOT EXISTS (
    SELECT 1
    FROM productos
    WHERE producto_id = p_producto_id
  ) THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El producto no existe.';
  END IF;
  DELETE FROM productos WHERE producto_id = p_producto_id;
  select concat('Reg.No.',convert(p_producto_id,char),' eliminado') as Registro_Eliminado;
END
```

## VISTAS

### ■ Crea una vista de los clientes con informacion distribuida en 5 tablas

```
CREATE
  ALGORITHM = UNDEFINED
  DEFINER = `root`@`localhost`
  SQL SECURITY DEFINER
VIEW `v_listado_clientes_consumo` AS
  SELECT
    `c`.`cliente_id` AS `cliente_id`,
    `p`.`nombre` AS `nombre`,
    `p`.`apellidos` AS `apellidos`,
    CALCULAR_EDAD(`p`.`fnacimiento`) AS `Edad`,
    REPLACE(FORMAT(`p`.`celular`, '#-###-###-###'),
      ',',
      '-') AS `celular`,
    `ci`.`ciudad` AS `ciudad`,
    `pa`.`pais` AS `pais`,
    FORMAT(CALCULAR_CONSUMO(`c`.`cliente_id`),
      '###,###.##') AS `consumo`
  FROM
    ((((`clientes` `c`
    JOIN `personas` `p` ON ((`c`.`persona_id` = `p`.`persona_id`)))
    JOIN `direcciones` `d` ON ((`d`.`direccion_id` = `p`.`direccion_id`)))
    JOIN `paises` `pa` ON ((`pa`.`pais_id` = `d`.`pais_id`)))
    JOIN `ciudades` `ci` ON ((`ci`.`ciudad_id` = `d`.`ciudad_id`)))
  ORDER BY `c`.`cliente_id`
```

cliente_id	nombre	apellidos	Edad	celular	ciudad	pais	consumo
1	Irena	Farge	16	4-291-899-366	Yakarta	Indonesia	0
2	Lenna	Tarquino	35	7-756-821-574	Brasilia	Brazil	0
3	Rees	Tilio	2	1-548-812-210	Brasilia	Brazil	41,114
4	Terra	Levane	23	8-418-917-925	Yakarta	Indonesia	153,438
5	Fidelity	Vasyaev	44	7-674-661-573	Brasilia	Brazil	134,579



- Crea una vista de los productos, stock y precio venta ordenado por proveedor

```
CREATE
ALGORITHM = UNDEFINED
DEFINER = `root`@`localhost`
SQL SECURITY DEFINER
VIEW `v_productos_proveedores` AS
SELECT
    `p`.`producto_id` AS `producto_id`,
    `p`.`nombre` AS `nombre`,
    `p`.`precioVenta` AS `precioVenta`,
    `p`.`cantidad` AS `cantidad`,
    `p`.`fechaCompra` AS `fechaCompra`,
    `pr`.`proveedor_id` AS `proveedor_id`,
    `pr`.`descripcion` AS `proveedor`
FROM
    ((`productos` `p`
    JOIN `productos_has_proveedores` `php` ON ((`p`.`producto_id` = `php`.`productos_producto_id`)))
    JOIN `proveedores` `pr` ON ((`php`.`proveedores_proveedor_id` = `pr`.`proveedor_id`)))
```

producto_id	nombre	precioVenta	cantidad	fechaCompra	proveedor_id	proveedor
30	Soup - Campbells Chili	56509.81	13.00	2023-05-11	1	Meemm
51	Sauce - Gravy, Au Jus, Mix	2242.98	58.00	2024-02-24	1	Meemm
74	Ham - Virginia	17402.42	81.00	2023-12-22	1	Meemm
132	Chickhen - Chicken Phyllo	81705.51	55.00	2023-07-06	1	Meemm
185	Easy Off Oven Cleaner	14919.63	95.00	2023-05-24	1	Meemm
199	Vinegar - Balsamic, White	75368.36	40.00	2023-10-10	1	Meemm

- Vista de los Servicios con sus tratamientos, costos, descripciones y precios de venta

```
CREATE
ALGORITHM = UNDEFINED
DEFINER = `root`@`localhost`
SQL SECURITY DEFINER
VIEW `v_servicios_tratamientos` AS
SELECT
    `s`.`servicio_id` AS `servicio_id`,
    `s`.`nombre` AS `servicio`,
    `s`.`descripcion` AS `descripcion`,
    `t`.`tratamiento_id` AS `tratamiento_id`,
    `t`.`nombre` AS `tratamiento`,
    `t`.`precioCosto` AS `precioCosto`,
    `t`.`precioVenta` AS `precioVenta`
FROM
    (`servicios` `s`
    LEFT JOIN `tratamientos` `t` ON ((`s`.`servicio_id` = `t`.`servicio_id`)))
```



servicio_id	servicio	descripcion	tratamiento_id	tratamiento	precioCosto	precioVent
1	Pasta - Lasagna Noodle, Frozen	vel augue vestibulum rutrum ru...	3	rutrum ac	16954.58	48918.78
1	Pasta - Lasagna Noodle, Frozen	vel augue vestibulum rutrum ru...	17	et	743.00	90204.49
1	Pasta - Lasagna Noodle, Frozen	vel augue vestibulum rutrum ru...	30	amet consectetur	49381.78	26519.94
2	Grouper - Fresh	eget nunc donec quis orci eget	1	tincidunt eu	50117.76	2882.94
3	Pork - Caul Fat	quam a odio in hac habitasse pl...	5	nulla	81900.65	69486.98
3	Pork - Caul Fat	quam a odio in hac habitasse pl...	18	sit amet	80714.13	78057.92
3	Pork - Caul Fat	quam a odio in hac habitasse pl...	20	nulla	57999.92	33069.98
4	Beef - Striploin	aliquam augue quam sollicitudi...	2	dis parturient	20330.41	7788.84
4	Beef - Striploin	aliquam augue quam sollicitudi...	16	quisque	41554.53	24209.23

## TRIGGERS

### ■ insert\_producto\_trigger INSERT AFTER

BEGIN

```
INSERT INTO `auditoria_productos`
(usuario, fecha_hora, accion, instruccion, id_producto, nombre_producto, precio_venta,precio_costo)
VALUES
(USER(), NOW(), "INSERT", "INSERT INTO productos VALUES (NULL, NEW.nombre,
NEW.precio_venta,NEW.precio_costo)", NEW.producto_id, NEW.nombre, NEW.precioventa,NEW.preciocosto);
END
```

### ■ update\_producto\_trigger UPDATE BEFORE

BEGIN

```
INSERT INTO `auditoria_productos`
(usuario, fecha_hora, accion, instruccion, id_producto, nombre_producto, precio_costo,precio_venta)
VALUES
(USER(), NOW(), 'UPDATE', CONCAT('UPDATE productos SET nombre = ', NEW.nombre, ', precio_costo = ',
convert(NEW.preciocosto,char), ', precio_venta = ', convert(NEW.precioventa,char), ' WHERE id = ',
convert(NEW.producto_id,char)), OLD.producto_id, OLD.nombre, OLD.preciocosto,OLD.precioventa);
END
```

### ■ delete\_producto\_trigger DELETE AFTER

BEGIN

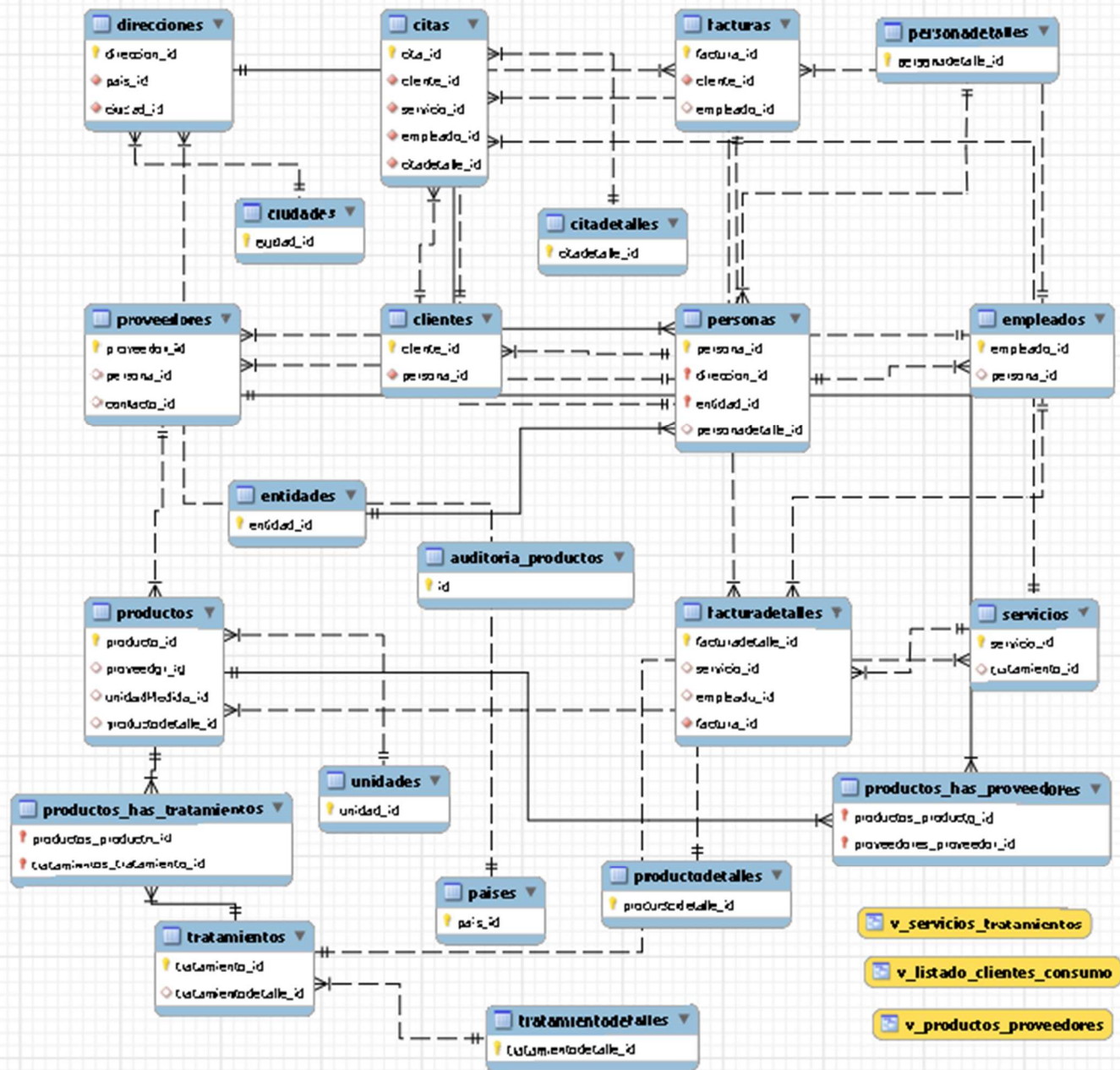
```
INSERT INTO `auditoria_productos`
(usuario, fecha_hora, accion, instruccion, id_producto, nombre_producto, precio_costo,precio_venta)
VALUES
(USER(), NOW(), 'DELETE', CONCAT('DELETE FROM productos WHERE id = ', convert(OLD.producto_id,char)),
OLD.producto_id, OLD.nombre, OLD.preciocosto,OLD.precioventa);
END
```



# Proyecto MySQL

Ronnie Alvarez  
Modelo Entidad  
Relación

Mel Coquette Beauty & Spa





## Detalles de Columnas y Claves de las Tablas

Table	Field	DataType	KeyType	Entity
auditoria_productos	id	int	PK	
auditoria_productos	usuario	varchar		
auditoria_productos	fecha_hora	datetime		
auditoria_productos	accion	varchar		
auditoria_productos	instruccion	text		
auditoria_productos	id_producto	int		
auditoria_productos	nombre_producto	varchar		
auditoria_productos	precio_costo	decimal		
auditoria_productos	precio_venta	decimal		
auditoria_productos	created_at	timestamp		
auditoria_productos	updated_at	timestamp		
citadetalles	citadetalles_id	int	PK	
citadetalles	citadetalles	text		
citadetalles	created_at	timestamp		
citadetalles	updated_at	timestamp		
citas	cita_id	int	PK	
citas	cliente_id	int	FK	clientes
citas	servicio_id	int	FK	servicios
citas	empleado_id	int	FK	empleados
citas	fecha_servicio	date		
citas	hora_servicio	time		
citas	citadetalles_id	int	FK	citadetalles
citas	created_at	timestamp		
citas	updated_at	timestamp		
ciudades	ciudad_id	int	PK	
ciudades	ciudad	varchar		
ciudades	pais_id	int		
ciudades	created_at	timestamp		
ciudades	updated_at	timestamp		
clientes	cliente_id	int	PK	
clientes	descripcion	varchar		
clientes	persona_id	int	FK	personas
clientes	created_at	timestamp		
clientes	updated_at	timestamp		
direcciones	direccion_id	int	PK	
direcciones	dir_detallada	text		
direcciones	pais_id	int	FK	países
direcciones	ciudad_id	int	FK	ciudades
direcciones	created_at	timestamp		
direcciones	updated_at	timestamp		
empleados	empleado_id	int	PK	
empleados	persona_id	int	FK	personas
empleados	tipo_contrato	varchar		
empleados	salario	decimal		



empleados	created_at	timestamp		
empleados	updated_at	timestamp		
entidades	entidad_id	int	PK	
entidades	nombre	varchar		
entidades	abrev	varchar		
entidades	created_at	timestamp		
entidades	updated_at	timestamp		
facturadetalles	facturadetalle_id	int	PK	
facturadetalles	facturadetalle	text		
facturadetalles	servicio_id	int	FK	servicios
facturadetalles	empleado_id	int	FK	empleados
facturadetalles	factura_id	int	FK	facturas
facturadetalles	created_at	timestamp		
facturadetalles	updated_at	timestamp		
facturas	factura_id	int	PK	
facturas	cliente_id	int	FK	clientes
facturas	cita_id	int		
facturas	total_fact	decimal		
facturas	tipo_fact	varchar		
facturas	fecha_pago	date		
facturas	forma_pago	varchar		
facturas	moneda	varchar		
facturas	empleado_id	int	FK	empleados
facturas	created_at	timestamp		
facturas	updated_at	timestamp		
países	pais_id	int	PK	
países	pais_id	int	PK	
países	pais	varchar		
países	codigo_pais	varchar		
países	created_at	timestamp		
países	updated_at	timestamp		
personadetalles	personadetalle_id	int	PK	
personadetalles	personadetalle	text		
personadetalles	created_at	timestamp		
personadetalles	updated_at	timestamp		
personas	persona_id	int	PK	
personas	persona_id	int	PK	
personas	idPers	int	PK	
personas	nombre	varchar		
personas	nombre	varchar		
personas	apellidos	varchar		
personas	paisId	int		
personas	email	varchar		
personas	telefono	varchar		
personas	celular	varchar		
personas	direccion_id	int	PK	



personas	direccion_id	int	PK	direcciones
personas	fecha_inicial	date		
personas	fecha_final	date		
personas	entidad_id	int	PK	
personas	entidad_id	int	PK	entidades
personas	personadetalle_id	int	FK	personadetalles
personas	fnacimiento	date		
personas	created_at	timestamp		
personas	updated_at	timestamp		
productodetalles	productodetalle_id	int	PK	
productodetalles	productodetalle	text		
productodetalles	created_at	timestamp		
productodetalles	updated_at	timestamp		
productos	producto_id	int	PK	
productos	nombre	varchar		
productos	precioCosto	decimal		
productos	precioVenta	decimal		
productos	cantidad	decimal		
productos	fechaCompra	date		
productos	proveedor_id	int	FK	proveedores
productos	unidadMedida_id	int	FK	unidades
productos	productodetalle_id	int	FK	productodetalles
productos	created_at	timestamp		
productos	updated_at	timestamp		
productos_has_proveedores	productos_producto_id	int	PK	
productos_has_proveedores	productos_producto_id	int	PK	productos
productos_has_proveedores	proveedores_proveedor_id	int	PK	
productos_has_proveedores	proveedores_proveedor_id	int	PK	proveedores
productos_has_proveedores	created_at	timestamp		
productos_has_proveedores	updated_at	timestamp		
productos_has_tratamientos	productos_producto_id	int	PK	
productos_has_tratamientos	productos_producto_id	int	PK	productos
productos_has_tratamientos	tratamientos_tratamiento_id	int	PK	
productos_has_tratamientos	tratamientos_tratamiento_id	int	PK	tratamientos
productos_has_tratamientos	created_at	timestamp		
productos_has_tratamientos	updated_at	timestamp		
proveedores	proveedor_id	int	PK	
proveedores	persona_id	int	FK	personas
proveedores	contacto_id	int	FK	personas
proveedores	descripcion	varchar		
proveedores	created_at	timestamp		
proveedores	updated_at	timestamp		
servicios	servicio_id	int	PK	
servicios	nombre	varchar		
servicios	descripcion	text		
servicios	tratamiento_id	int	FK	tratamientos



servicios	created_at	timestamp		
servicios	updated_at	timestamp		
tratamientodetalles	tratamientodetalle_id	int	PK	
tratamientodetalles	tratamientodetalle	text		
tratamientodetalles	created_at	timestamp		
tratamientodetalles	updated_at	timestamp		
tratamientos	tratamiento_id	int	PK	
tratamientos	producto_id	int		
tratamientos	servicio_id	int	FK	
tratamientos	nombre	varchar		
tratamientos	precioCosto	decimal		
tratamientos	precioVenta	decimal		
tratamientos	cantidad	decimal		
tratamientos	fechaCompra	date		
tratamientos	tratamientodetalle_id	int	FK	tratamientodetalles
tratamientos	created_at	timestamp		
tratamientos	updated_at	timestamp		
unidades	unidad_id	int	PK	
unidades	unidad	varchar		
unidades	created_at	timestamp		
unidades	updated_at	timestamp		
v_listado_clientes_consumo	cliente_id	int		
v_listado_clientes_consumo	nombre	varchar		
v_listado_clientes_consumo	apellidos	varchar		
v_listado_clientes_consumo	Edad	int		
v_listado_clientes_consumo	celular	varchar		
v_listado_clientes_consumo	ciudad	varchar		
v_listado_clientes_consumo	pais	varchar		
v_listado_clientes_consumo	consumo	varchar		
v_productos_proveedores	producto_id	int		
v_productos_proveedores	nombre	varchar		
v_productos_proveedores	precioVenta	decimal		
v_productos_proveedores	cantidad	decimal		
v_productos_proveedores	fechaCompra	date		
v_productos_proveedores	proveedor_id	int		
v_productos_proveedores	proveedor	varchar		
v_servicios_tratamientos	servicio_id	int		
v_servicios_tratamientos	servicio	varchar		
v_servicios_tratamientos	descripcion	text		
v_servicios_tratamientos	tratamiento_id	int		
v_servicios_tratamientos	tratamiento	varchar		
v_servicios_tratamientos	precioCosto	decimal		
v_servicios_tratamientos	precioVenta	decimal		



# Relaciones entre Tablas

Table1	Col 1	Constraint	Table 2	Col 2
<b>citas</b>	cliente_id	citas_ibfk_1	clientes	cliente_id
<b>citas</b>	servicio_id	citas_ibfk_2	servicios	servicio_id
<b>citas</b>	empleado_id	citas_ibfk_3	empleados	empleado_id
<b>citas</b>	citadetalle_id	citas_ibfk_4	citadetalles	citadetalle_id
<b>clientes</b>	persona_id	clientes_ibfk_1	personas	persona_id
<b>direcciones</b>	pais_id	direcciones_ibfk_1	países	pais_id
<b>direcciones</b>	ciudad_id	direcciones_ibfk_2	ciudades	ciudad_id
<b>empleados</b>	persona_id	empleados_ibfk_1	personas	persona_id
<b>facturadetalles</b>	factura_id	facturadetalles_ibfk_1	facturas	factura_id
<b>facturadetalles</b>	servicio_id	facturadetalles_ibfk_2	servicios	servicio_id
<b>facturadetalles</b>	empleado_id	facturadetalles_ibfk_3	empleados	empleado_id
<b>facturas</b>	empleado_id	facturas_ibfk_1	empleados	empleado_id
<b>facturas</b>	cliente_id	facturas_ibfk_2	clientes	cliente_id
<b>personas</b>	personadetalle_id	personas_ibfk_1	personadetalles	personadetalle_id
<b>personas</b>	entidad_id	personas_ibfk_2	entidades	entidad_id
<b>personas</b>	direccion_id	personas_ibfk_3	direcciones	direccion_id
<b>productos</b>	unidadMedida_id	productos_ibfk_1	unidades	unidad_id
<b>productos</b>	proveedor_id	productos_ibfk_2	proveedores	proveedor_id
<b>productos</b>	productodetalle_id	productos_ibfk_3	productodetalles	productodetalle_id
<b>productos_has_proveedores</b>	proveedores_proveedor_id	productos_has_proveedores_ibfk_1	proveedores	proveedor_id
<b>productos_has_proveedores</b>	productos_producto_id	productos_has_proveedores_ibfk_2	productos	producto_id
<b>productos_has_tratamientos</b>	productos_producto_id	productos_has_tratamientos_ibfk_1	productos	producto_id
<b>productos_has_tratamientos</b>	tratamientos_tratamiento_id	productos_has_tratamientos_ibfk_2	tratamientos	tratamiento_id
<b>proveedores</b>	persona_id	proveedores_ibfk_1	personas	persona_id
<b>proveedores</b>	contacto_id	proveedores_ibfk_2	personas	persona_id
<b>servicios</b>	tratamiento_id	servicios_ibfk_1	tratamientos	tratamiento_id
<b>tratamientos</b>	tratamientodetalle_id	tratamientos_ibfk_1	tratamientodetalles	tratamientodetalle_id