

Optica Barba



Proyecto Final
Seminario De Solucion de Problemas de Bases de Datos
D10

Integrantes:

Barba Palacios Orlando Agustín

Código: 214798072

Gutiérrez Galán Rubén Alejandro

Código: 214798315

Índice:

Problemática:	3
Solución:	3
Objetivo:	3
Entrevista al cliente:	4
Requerimientos.	5
Requerimientos funcionales:	5
Requerimientos no funcionales:	5
Diagrama de Gantt:	6
Diagrama de entidad-relación:	7
Modelo Relacional:	8
Diccionario de datos:	9
Creación de Tablas:	17
Creación de Views:	19
Insert:	20
Delete:	21
select:	22
update:	23
Triggers:	24
Conclusiones:	25

Problemática:

El señor Gabriel Barba (optometrista) nos solicitó Digitalizar el registro de clientes y nota de venta, basados en las notas de venta físicas que utiliza en su negocio “Optica Barba”, “hay ocasiones en que tengo mucho trabajo por lo cual no guardo los registros en orden, esto me ha generado problemas debido a que muchas veces termino perdiendo notas, las cuales son muy importantes ya que tienen los datos de graduación y precio de los lentes que estoy vendiendo”.

El señor Barba nos menciona que se han presentado casos en los cuales ha tenido que vender lentes a precios más bajos de los que realmente están cotizados a un cliente, debido a que extravió la nota de venta de dicho cliente, por lo cual solicita que se le cree una base de datos en la cual el pueda almacenar los datos de cliente (nombre, teléfono, los datos de graduación del producto en venta “lentes”, precio y fecha de venta), de igual manera él desea poder modificar dichos registros “también quiero que se puedan modificar los datos, ya que tengo algunos clientes frecuentes a los cuales les vendo lentes cada cierto tiempo, y no quiero estar volviendo a hacer notas, si es posible poder modificar los graduación y los datos personales ”, como última petición quiere que se puedan eliminar los registros de los clientes (esto incluyendo los datos personales, datos de graduación y datos de venta que se le hayan realizado) debido a que considera que todos estos datos son “archivos muertos”.

Solución:

Se desarrollará una aplicación web con la cual el señor Gabriel Barba (Optometrista) podrá registrar clientes y notas de venta, almacenando estos datos en una base de datos relacional para un mejor manejo y organización de dichas notas de venta.

Objetivo:

La aplicación web nos ofrecerá la opción de registrar datos de un cliente como su nombre y teléfono de contacto así como la graduación correspondiente además de los lentes que se le venderán.

Para el registro de una nota de venta se podrán realizar sobre un cliente existente y un cliente nuevo.

Las notas de venta van dirigidas a un cliente y a los lentes que se van a vender, así como su precio y fecha de la venta.

Cada que el optometrista quiera realizar una nueva venta a un cliente existente se le mostrarán las notas de venta pasadas de dicho cliente, por lo que podrá ver las cotizaciones pasadas de ese cliente para poder hacer una nueva cotización.

El optometrista podrá buscar y visualizar los datos de sus clientes por el nombre del cliente.

Para registrar un nuevo cliente este debe comprar unos lentes, de lo contrario no se registrara como cliente.

Entrevista al cliente:

- ¿Cuál es su nombre?
Gabriel Barba Cárdenas.
- ¿Qué servicio ofrece?
Soy un optometrista, hago exámenes de la vista además de adaptar lentes y pupilentes.
- ¿Cuál es la ubicación de su negocio?
Pablo Valdez #2613 colonia San Isidro, entre Abraham González y Lagunitas.
- ¿Podría darnos información básica de su negocio?
Adapto lentes y pupilentes, realizó exámenes de la vista además de también hacer reparaciones de lentes.
- ¿Qué características tiene en mente para su aplicación web?
Quiero un apartado para registrar a mis clientes y mis notas de venta, además de poder visualizar dichas notas de venta e información de mis clientes, debido a que el control de mis notas lo hago de manera física y suelen estar en desorden por lo que me quita tiempo buscar los datos de alguno de mis clientes.
- ¿Tiene algún formato específico para sus registros de clientes y notas de venta?
Si, de hecho utilizó un formato que cumple con ambos objetivos, en ese formato escribo los datos de mi cliente, su graduación, los lentes que le voy a adaptar y el precio que le di a dichos lentes, el costo varía dependiendo de las necesidades de mi cliente.

	OPTICA BARBA Gabriel Barba Cárdenas PABLO VALDEZ No. 2613, C.P 44750 S.L. GUADALAJARA JAL TEL: 33-20-40-37-50 Correo: gabeop@hotmail.com	FECHA DE EXPEDICIÓN DIA MES AÑO	
		FECHA DE ENTREGA DIA MES AÑO	
NOMBRE: _____			
TELEFONO: _____			
DOMICILIO: _____			
OJO DERECHO ESF CIL EJE	DP	O	OJO IZQUIERDO ESF CIL EJE
LEJOS			LEJOS
CERCA			CERCA
TIPO: _____ MATERIAL: _____ E: _____			
ARMAZON: _____ COLOR: _____ TAMAÑO: _____			
OBSERVACIONES: _____			
PRESENTA ESTA NOTA AL RECOGER SUS LENTES DESPUES DE 40 DIAS NO NOS HACEMOS RESPONSABLES POR SU MERCANCIA Y SU ANTICIPO		PRECIO TOTAL ANTICIPO _____ RESTA _____	

- ¿Usa algún método u operación para saber el precio que le dará al cliente para la venta de unos lentes?
No, suelen influir muchos aspectos al momento de dar un precio para unos lentes, por lo que no es necesario que automaticen esa parte, con que pueda tener un registro similar a mis notas de venta actuales estaré satisfecho.
- ¿Requiere algo en particular para hacer una nueva venta a un cliente? O basta con el formato de nota de venta.
Para vender unos lentes a un nuevo cliente solo necesito mi formato de nota de venta, pero para realizar una nueva nota de venta a un cliente existente me gustaría visualizar las notas de venta anteriores de ese cliente al mismo tiempo que realizó la nueva nota.
- ¿Desea mantener su formato de nota de venta tal cual está?
No, de hecho estas notas de venta que guardaré en la aplicación están destinadas para mi propia consulta y no para el cliente por lo que no necesitaré los datos de la óptica que están al comienzo, tampoco la fecha de entrega, la leyenda de la

garantía, el anticipo y la resta, básicamente solo quiero poder registrar los datos del cliente, su graduación, los lentes que comprara y el costo total de la venta.

- [¿Tiene un tiempo límite y mínimo para la entrega de la página que solicita?](#)
Por el momento no me urge la aplicación, sin embargo me gustaría tenerla antes del próximo año.
- [¿Podría proporcionarnos algún medio para comunicarnos con usted?](#)
Mi número de teléfono es 33-2040-3750 y mi correo es: gabeop@hotmail.com.

Requerimientos.

Requerimientos funcionales:

- Para registrar un nuevo cliente se deben registrar sus datos, graduación, lentes a comprar y precio de venta a la vez.
- Los datos básicos de un cliente deberán ser su nombre y teléfono.
- Para registrar la graduación es necesario registrar los siguientes datos: esférico Izquierdo de lejos, esférico izquierdo de cerca, esférico derecho de lejos, esférico derecho de cerca, cilindro izquierdo de lejos, cilindro izquierdo de cerca, cilindro derecho de Lejos, cilindro derecho de cerca, eje izquierdo de lejos, eje izquierdo de cerca, eje derecho de lejos, eje derecho de cerca, distancia de pupila y altura.
- Para registrar unos lentes se necesitan los siguientes datos: tipo, material, información extra, armazón, color, tamaño y observaciones.
- Para registrar una venta solo se necesitará registrar la fecha de realización y precio total de la venta.
- Mostrar las notas de ventas anteriores de un cliente al que se le registrara una nueva venta.
- Poder buscar un cliente por su nombre y mostrar sus datos y graduación.
- Poder modificar los datos de un cliente y de la graduación de ese mismo cliente.
- Poder eliminar un cliente y al hacerlo borrar todos los datos e información relacionada a él.
- Regalar un estuche de lentes con cada compra.

Requerimientos no funcionales:

- Para guardar los datos de una nota de venta dividiremos el registro en clientes, graduación, lentes,y ventas.
- Cada cliente tiene una sola graduación la cual está ligada a este por un identificador único de cliente,
- cada cliente puede realizar varias ventas, las cuales están ligadas a este por un identificador único de cliente.
- Las ventas también tienen unos lentes por venta, estos están ligados por un identificador de lentes único de lentes.
- Las ventas guardan en automático la fecha del día en que se realizan, sin necesidad de ponerla el cliente.

- La búsqueda de un cliente se realiza buscando coincidencia en el nombre basado en una cadena de caracteres.
- La eliminación se apoya en la eliminación en cascada sql debido a que al ser un cliente la tabla independiente y graduación y venta tablas dependientes de esta por uso de llave foránea, cuando el cliente es eliminado también las ventas y su graduación son eliminadas.
- Nos apoyamos en un trigger para la eliminación de lentes relacionados a las ventas que han sido eliminadas.
- Usar un trigger que decremente en 1 la cantidad de estuches de lentes cada que se realiza un registro de venta.

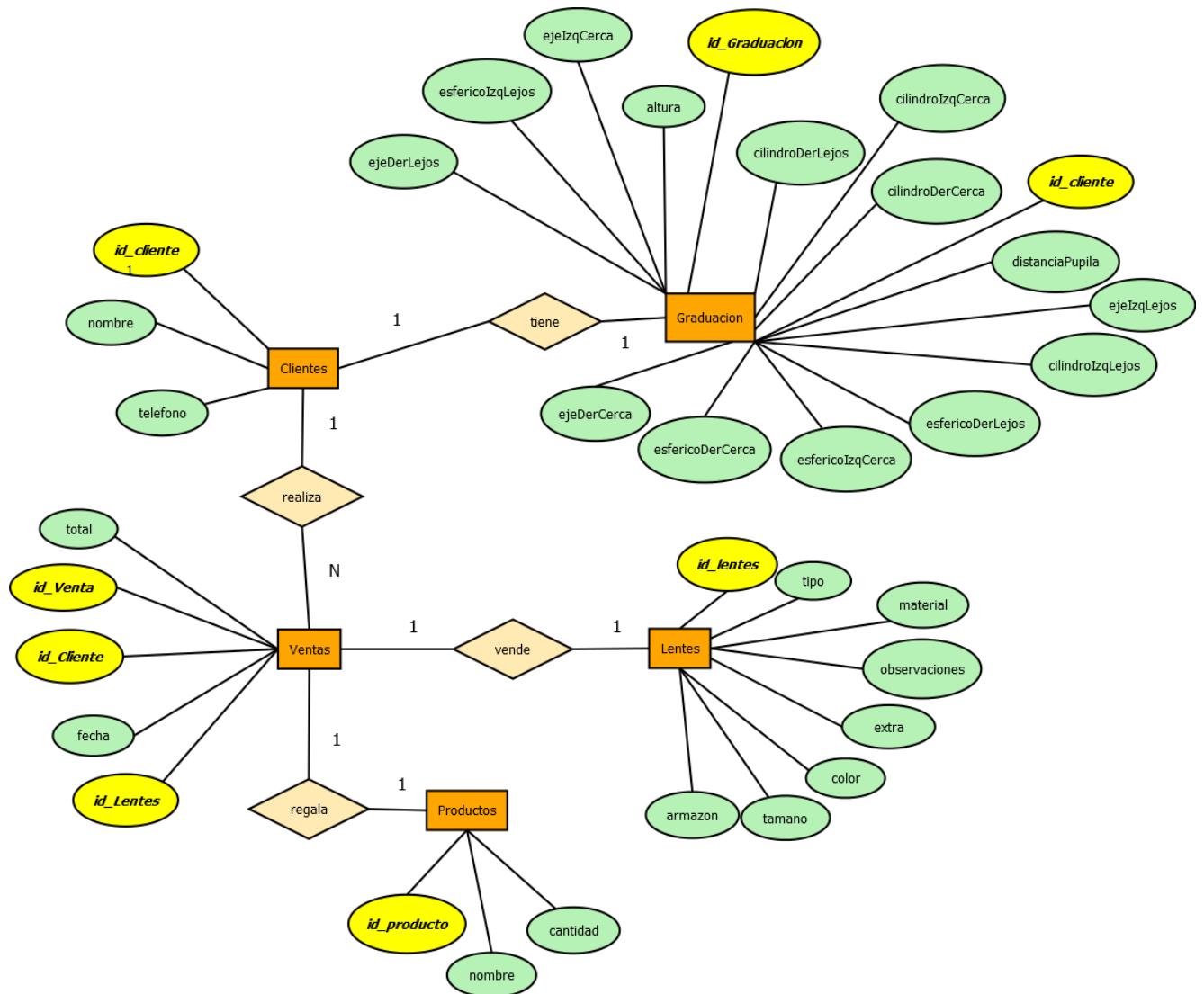
Diagrama de Gantt:

Actividad	Mayo				
	10-05-2021/13-05-2021	17/05/2021	20-05-2021/23-05-2021	25/05/2021	28-05-2021/30-05-2021
Planteamiento de la problemática					
Entrevista (Gabriel Barba)					
Planteamiento de la Solución					
Creación de Tablas para la BD					
Creación de Funciones para Triggers					
Desarrollo de Triggers					
Desarrollo de Estructura de la Pagina Web					
Funciones de la pagina web					
Diseño de la pagina web (estilo)					
Documentación					
Vídeo demostrativo					

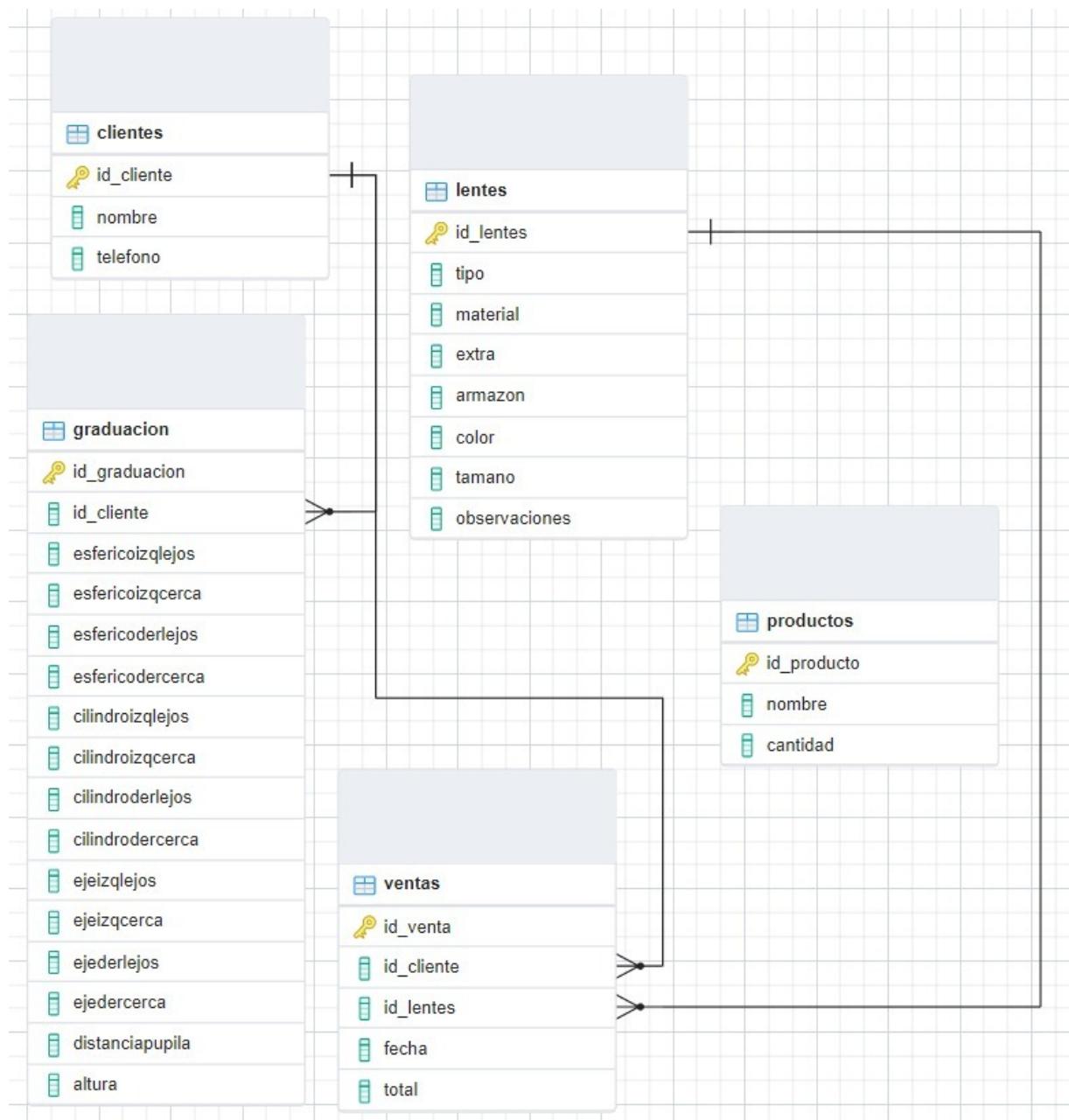
Actividad	Junio				
	01-06-2021/04-06-2021	05-06-2021/07-05-2021	08-06-2021/09-06-2021	10-06-2021/12-06-2021	13-06-2021/15-06-2021
Planteamiento de la problemática					
Entrevista (Gabriel Barba)					
Planteamiento de la Solución					
Creación de Tablas para la BD					
Creación de Funciones para Triggers					
Desarrollo de Triggers					
Desarrollo de Estructura de la Pagina Web					
Funciones de la pagina web					
Diseño de la pagina web (estilo)					
Documentación					
Vídeo demostrativo					

Orlando Barba		
Ruben Gutierrez		
Ambos		

Diagrama de entidad-relación:



Modelo Relacional:



Diccionario de datos:

Logotipo	Nombre de la empresa: Optica Barba Diccionario de Datos
----------	--

Nombre de la Tabla	Clientes	Fecha de creación	25/05/2021
Descripción:	Almacena los datos personales del cliente		

Nombre de Campo	Tipo de Dato	Tamaño	Valor por default	Fórmula	Valor nulo	Descripción
id_Cliente	serial	small	0	serial	no	identificador del cliente
nombre	char	50	desconocido		si	nombre del cliente
telefono	varchar	11	desconocido		si	teléfono del cliente

Relaciones: Graduacion, Ventas, Lentes	Campos clave: id_Cliente
--	--------------------------

Logotipo	Nombre de la empresa: Optica Barba Diccionario de Datos
----------	--

Nombre de la Tabla	Productos	Fecha de creación	25/05/2021
Descripción:	contiene el registro de almacén de estuches (producto extra)		

Nombre de Campo	Tipo de Dato	Tamaño	Valor por default	Fórmula	Valor nulo	Descripción
id_Producto	serial	small	0	serial	no	identificador de producto extra
nombre	varchar	50	desconocido		no	nombre del producto extra
cantidad	int	small	desconocido		no	cantidad de producto extra

Relaciones:	Campos clave: id_producto
-------------	---------------------------

Logotipo	Nombre de la empresa: Optica Barba Diccionario de Datos
----------	--

Nombre de la Tabla	Lentes	Fecha de creación	25/05/2021
Descripción:	Almacena los datos del producto lentes (incluye la información de graduacion)		

Nombre de Campo	Tipo de Dato	Tamaño	Valor por default	Fórmula	Valor nulo	Descripción
id_lentes	serial	small	0	serial	no	identificador del producto lentes
tipo	varchar	30	desconocido		no	tipo de lentes (para distancia largas, cortas o fotocromáticos)
material	varchar	30	desconocido		no	indica la aleación de la cual está hecho el producto (lentes)
extra	varchar	30	desconocido		no	agregados al producto (características especificadas)

						por el comprador)
amazon	varchar	30	desconocido		no	tipo de amazon que se utilizara en los lentes (plastico, metalicos)
color	varchar	30	desconocido		si	color del armazón
tamano	varchar	30	0.00		no	tamaño del armazón
observaciones	varchar	100			si	criterios de evaluación establecidos por el optometrista

Relaciones: Ventas, Clientes	Campos clave: id_lentes
------------------------------	-------------------------

Logotipo	Nombre de la empresa: Optica Barba Diccionario de Datos
----------	--

Nombre de la Tabla	Ventas	Fecha de creación	25/05/2021
Descripción:	Almacena los registros de venta del producto lentes		

Nombre de Campo	Tipo de Dato	Tamaño	Valor por default	Fórmula	Valor nulo	Descripción
id_Venta	serial	small	0	serial	no	identificador de venta
id_Cliente	smallint	small	desconocido		no	identificador del cliente al cual se le realiza la venta
id_Lentes	smallint	small	desconocido		no	identificador del producto lentes vendido
fecha	date		00/00/00		no	fecha en la que se realizó la venta

total	int		0.00		no	precio total por el cual se realizó la venta del producto lentes
-------	-----	--	------	--	----	--

Relaciones: Clientes, Lentes, Graduacion	Campos clave: id_Venta (primary key), id_Cliente (foreign key), id_Lentes (foreign key)
--	---

Logotipo	Nombre de la empresa: Optica Barba Diccionario de Datos
----------	--

Nombre de la Tabla	Graduacion	Fecha de creación	25/05/2021
Descripción:	Datos recopilados para la adaptación del producto lentes		

Nombre de Campo	Tipo de Dato	Tamaño	Valor por default	Fórmula	Valor nulo	Descripción
-----------------	--------------	--------	-------------------	---------	------------	-------------

id_Graduación	serial	small	0	serial	no	identificador de la graduación usada para el producto lentes
id_cliente	smallint	small	0		no	identificador del cliente al cual se le realizó el estudio de graduación
esfericolzqLejos	varchar	10	0		no	esférico izquierdo de lejos
esfericolzqCerca	varchar	10	0		no	esférico izquierdo de cerca
esfericoDerLejos	varchar	10	0		no	esferico derecho de lejos
esfericoDerCerca	varchar	10	0		no	esferico derecho de cerca
cilindrolzqLejos	varchar	10	0		no	cilindro izquierdo de lejos

cilindrolzqCerca	varchar	10	0		no	cilindro izquierdo de cerca
cilindroDerLejos	varchar	10	0		no	cilindro derecho de lejos
cilindroDerCerca	varchar	10	0		no	cilindro derecho de cerca
ejeIzqLejos	varchar	10	0		no	eje izquierdo de lejos
ejeIzqCerca	varchar	10	0		no	eje izquierdo de cerca
ejeDerLejos	varchar	10	0		no	eje derecho de lejos
ejeDerCerca	varchar	10	0		no	eje derecho de cerca

distanciaPupila	varchar	10	0		no	distancia de la pupila
altura	varchar	10	0		no	altura de la pupila

Relaciones: Clientes, Ventas	Campos clave: id_Graduacion (primary key), id_cliente (foreign key)
------------------------------	---

Creación de Tablas:

Tabla Clientes:

```
OpticaBarba/postgres@PostgreSQL 13 
Query Editor Query History
1 create table Clientes
2 (
3     id_Cliente serial,
4     nombre varchar(50),
5     telefono varchar(11),
6     primary key(id_Cliente)
7 );
```

Messages Data Output

```
CREATE TABLE
```

Query returned successfully in 359 msec.

Tabla Productos:

```
OpticaBarba/postgres@PostgreSQL 13 
Query Editor Query History
1 create table Productos
2 (
3     id_Producto serial,
4     nombre varchar(50),
5     cantidad int,
6     primary key(id_Producto)
7 );
```

Messages Data Output

```
CREATE TABLE
```

Query returned successfully in 78 msec.

Tabla Lentes:

```
OpticaBarba/postgres@PostgreSQL 13 
Query Editor Query History
1 create table Lentes
2 (
3     id_Lentes serial,
4     tipo varchar(30),
5     material varchar(30),
6     extra varchar(30),
7     amazon varchar(30),
8     color varchar(30),
9     tamano varchar(30),
10    observaciones varchar(100),
11    primary key(id_Lentes)
12 );
```

Messages Data Output

```
CREATE TABLE
```

Query returned successfully in 81 msec.

Tabla Ventas:

```
OpticaBarba/postgres@PostgreSQL 13 
Query Editor Query History
1 create table Ventas
2 (
3     id_Venta serial,
4     id_Cliente smallint not null,
5     id_Lentes smallint not null,
6     fecha date,
7     total int,
8     CONSTRAINT fk_Clientes FOREIGN KEY (id_Cliente) references Clientes(id_Cliente),
9     CONSTRAINT fk_Lentes FOREIGN KEY (id_Lentes) references Lentes(id_Lentes),
10    primary key(id_Venta)
11 );
```

Messages Data Output

```
CREATE TABLE
```

Query returned successfully in 125 msec.

Tabla Graduacion:

```
OpticaBarba/postgres@PostgreSQL 13 
Query Editor Query History
1 create table Graduacion
2 (
3     id_Graduacion serial,
4     id_Cliente smallint not null,
5     esfericoIzqLejos varchar(10),
6     esfericoIzqCerca varchar(10),
7     esfericoDerLejos varchar(10),
8     esfericoDerCerca varchar(10),
9     cilindroIzqLejos varchar(10),
10    cilindroIzqCerca varchar(10),
11    cilindroDerLejos varchar(10),
12    cilindroDerCerca varchar(10),
13    ejeIzqLejos varchar(10),
14    ejeIzqCerca varchar(10),
15    ejeDerLejos varchar(10),
16    ejeDerCerca varchar(10),
17    distanciaPupila varchar(10),
18    altura varchar(10),
19    CONSTRAINT fk_Clientes FOREIGN KEY (id_Cliente) references Clientes(id_Cliente),
20    primary key(id_Graduacion)
21 );
```

Creación de Views:

The screenshot shows a PostgreSQL query editor interface. The title bar says "OpticaBarba/postgres@PostgreSQL 13". Below the title bar, there are two tabs: "Query Editor" (which is selected) and "Query History". The main area contains the following SQL code:

```
1 create view view_cliente_graduacion as
2 select
3 clientes.nombre,
4 clientes.telefono,
5 graduacion.esfericoIzqLejos,
6 graduacion.esfericoIzqCerca,
7 graduacion.esfericoDerLejos,
8 graduacion.esfericoDerCerca,
9 graduacion.cilindroIzqLejos,
10 graduacion.cilindroIzqCerca,
11 graduacion.cilindroDerLejos,
12 graduacion.cilindroDerCerca,
13 graduacion.ejeIzqLejos,
14 graduacion.ejeIzqCerca,
15 graduacion.ejeDerLejos,
16 graduacion.ejeDerCerca,
17 graduacion.distanciapupila,
18 graduacion.altura
19 from clientes, graduacion
20 where clientes.id_cliente = graduacion.id_cliente;
```

Nos permitirá visualizar de mejor manera los datos de graduación que almacenemos de los clientes que vayamos registrando según se realicen ventas.



OpticaBarba/postgres@PostgreSQL 13 ▾

Query Editor Query History

```
1 create view view_NotaDeVenta as select
2 clientes.nombre,
3 clientes.telefono,
4 graduacion.esfericoIzqLejos,
5 graduacion.esfericoIzqCerca,
6 graduacion.esfericoDerLejos,
7 graduacion.esfericoDerCerca,
8 graduacion.cilindroIzqLejos,
9 graduacion.cilindroIzqCerca,
10 graduacion.cilindroDerLejos,
11 graduacion.cilindroDerCerca,
12 graduacion.ejeIzqLejos,
13 graduacion.ejeIzqCerca,
14 graduacion.ejeDerLejos,
15 graduacion.ejeDerCerca,
16 graduacion.distanciaPupila,
17 graduacion.altura,
18 lentes.tipo,
19 lentes.material,
20 lentes.extra,
21 lentes.amazon,
22 lentes.color,
23 lentes.tamaño,
24 lentes.observaciones,
25 ventas.fecha,
26 ventas.total
27 from clientes, graduacion, lentes, ventas
28 where clientes.id_cliente = graduacion.id_cliente
29 and clientes.id_cliente = ventas.id_cliente
30 and ventas.id_lentes = lentes.id_lentes
```

Cuenta con toda la información del cliente y la graduación de la cual dispone dicho cliente.

Insert:



optica_barba/postgres@PostgreSQL 13 ▾

Query Editor Query History

```
1 insert into clientes values(50,'Pepe Gonzalez','3318201457');
2
3
```

	id_cliente [PK] integer	nombre character varying (50)	telefono character varying (11)
1	33	Gabriel Barba	331713055
2	41	Karla	3336268605
3	44	Iker Daniel	214798072
4	32	Orlando Barba	3331509512
5	45	Israel Castellanos	123456789
6	50	Pepe Gonzalez	3318201457

nuevo cliente agregado

```
insert into graduacion values(9,50,'b','c','c','c','d','12','34','7','8','9','23','3','90');
```

Data Output Explain Messages Notifications						
	id_graduacion [PK] integer	id_cliente smallint	esfericoizqlejos character varying (10)	esfericoizqcerca character varying (10)	esfericoderlejos character varying (10)	
1	9	50	b	c	c	c

Data Output Explain Messages Notifications						
	esfericodercerca character varying (10)	cilindroizqlejos character varying (10)	cilindroizqcerca character varying (10)	cilindroderlejos character varying (10)	cilindrodercerca character varying (10)	ejeizqlejos character varying (10)
c	c	d	12	34	7	8

ejederlejos character varying (10)	ejedercerca character varying (10)	distanciapupila character varying (10)	altura character varying (10)
9	23	3	90

se agregó nueva graduación.

Delete:

```
delete from graduacion where id_graduacion=9;
```

Output Explain Messages Notifications			
STATE 1			
Query returned successfully in 56 msec.			

Data Output Explain Messages Notifications

	id_graduacion [PK] integer	id_cliente smallint	esfericoizqlejos character varying (10)	esfericoizqcerca character varying (10)	esfericoderlejos character varying (10)	esfericodercerca character varying (10)	cilindroizqlejos character varying (10)	cilind char
1	4	33	7	10	1	4	8	11
2	13	41	I	C	a	f	z	v
3	16	44	asereje	12	ik	ik	1e+	3
4	3	32	f	f	f	f	f	f
5	17	45	is+	is+	is+			

✓ Successfully run. Total query runtime: 56 msec. 5 rows affected.

se eliminó el registro con el id_graduacion 9.

```
11
12 delete from clientes where id_cliente=50;
```

Data Output Explain Messages Notifications

DELETE 1

Query returned successfully in 62 msec.

Data Output Explain Messages Notifications

	id_cliente [PK] integer	nombre character varying (50)	telefono character varying (11)
1	33	Gabriel Barba	331713055
2	41	Karla	3336268605
3	44	Iker Daniel	214798072
4	32	Orlando Barba	3331509512
5	45	Israel Castellanos	123456789

se eliminó el cliente cuyo id_cliente es 50.

select:

```
6
7 select*from clientes where id_cliente=33;
8
```

	id_cliente [PK] integer	nombre character varying (50)	telefono character varying (11)
1	33	Gabriel Barba	331713055

seleccionamos de la tabla clientes todos los atributos pero solo tomamos los pertenecientes al registro cuyo id_cliente es 33.

```
select altura as altura, id_graduacion as id_de_graduacion, id_cliente as cliente from graduacion where id_graduacion=13;
```

Data Output				Explain	Messages	Notifications
	altura character varying (10)	id_de_graduacion integer	cliente smallint			
1	k	13	41			

mostramos sólo la altura, el id_graduacion y el id_cliente de la tabla graduacion que pertenecen al registro cuyo id_graduacion es 13.

update:

Data Output				Explain	Messages	Notifications
	id_cliente [PK] integer	nombre character varying (50)	telefono character varying (11)			
1	33	Gabriel Barba	331713055			
2	41	Karla	3336268605			
3	44	Iker Daniel	214798072			
4	32	Orlando Barba	3331509512			
5	45	Israel Castellanos	123456789			

```
update clientes set nombre='Gabriel Barba Montes' where nombre='Gabriel Barba';
```

	id_cliente [PK] integer	nombre character varying (50)	telefono character varying (11)
1	41	Karla	3336268605
2	44	Iker Daniel	214798072
3	32	Orlando Barba	3331509512
4	45	Israel Castellanos	123456789
5	33	Gabriel Barba Montes	331713055

modificamos el nombre del registro Gabriel Barba por Gabriel Barba Montes.

```
update clientes set nombre='Karla Palacios' where nombre='Karla';
```

	id_cliente [PK] integer	nombre character varying (50)	telefono character varying (11)
1	44	Iker Daniel	214798072
2	32	Orlando Barba	3331509512
3	45	Israel Castellanos	123456789
4	33	Gabriel Barba Montes	331713055
5	41	Karla Palacios	3336268605

modificamos el nombre del registro Karla por Karla Palacios.

Triggers:

Función bajarEstuche():

Query Editor Query History

```

1  create function bajarEstuche() returns Trigger
2  as
3  $$
4  begin
5
6  update productos set cantidad = (cantidad - 1) where nombre = 'Estuches';
7  return new;
8
9  end
10 $$ 
11 language plpgsql;
```

Función que nos permitirá eliminar un producto (Estuche) de nuestro stock, esta función será utilizada cada vez que se realice una venta (por cada par de lentes vendidos se entrega un estuche como producto extra “regalo”).

Creación del Trigger estucheRegalo_ventas:

Query Editor Query History

```

1  create trigger estucheRegalo_ventas after insert on ventas
2  for each row
3  execute procedure bajarEstuche();
4  |
```

este trigger se implementará despues de que se agregue un nuevo registro a nuestra tabla ventas, este mandara llamar a la función bajarEstuche anteriormente descrita permitiéndonos realizar tanto un eliminacion como una inserción (eliminación de la tabla Productos e inserción en la tabla ventas).

Creación de la función eliminarLentes:

```
OpticaBarba/postgres@PostgreSQL 13 ▾
Query Editor Query History
1 create function eliminarLentes() returns Trigger
2 as
3 $$begin
4
5 delete from lentes where id_lentes = old.id_lentes;
6 return new;
7
8 end
9 $$language plpgsql;
```

Función que nos permite eliminar un registro de la tabla Lentes por medio del id que hemos solicitado (será utilizada cuando realicemos una venta).

Creación del Trigger eliminarLentes_venta:

```
optica_barba/postgres@PostgreSQL 13 ▾
Query Editor Query History
1 create trigger eliminarLentes_venta after delete on ventas
2 for each row
3 execute procedure eliminarLentes();
```

Este Trigger se activará después de que se realice una venta, eliminando los datos del producto en cuestión “lentes” por medio del llamado a la función eliminarLentes.

Conclusiones:

Ruben:

Gracias al desarrollo de este proyecto pusimos en práctica lo aprendido durante el curso de Seminario de Solución de Problemas de Bases de Datos, este proyecto nos permitió experimentar de manera más práctica lo que es el mundo laboral real, nuestra base de datos fue desarrollada con el fin de generar un beneficio para nuestro cliente el señor Gabriel Barba, después de la entrevista realizada comprendimos que teníamos una gran oportunidad, al aceptar el trabajo no solo tomamos la responsabilidad de entregar un producto funcional sino que en parte tomamos la responsabilidad disminuir la carga de trabajo que el señor Barba tenía, generar un acercamiento hacia el uso nuevas tecnologías (ayudarlo a pasar del uso de almacenamiento de información por métodos convencionales “papel”, a métodos más eficientes “de manera digital”), en lo personal consideramos que

nuestra labor como ingenieros en el desarrollo de nuevas tecnologías es acercar cada vez más a los ciudadanos a perder el “miedo o rechazo” hacia el uso herramientas digitales.

Orlando:

Este proyecto no solo sirvió para aplicar y demostrar los conocimientos adquiridos en la clase de seminario de base de datos, sino también para dar base a una solución computacional para mi padre que fue el que nos propuso ayudarlo con su problema de registro de notas de venta para así tenerlas más organizadas para su uso y consulta, pude aplicar mis conocimientos previos en desarrollo web para realizar con éxito este proyecto, si bien es cierto que para lo que deseamos programar hubiese sido mejor una aplicación de escritorio y no una herramienta web, pero debido a la facilidad de creación de una interfaz gráfica, de una conexión fácil a la base de datos y que tanto Rubén como yo teníamos el conocimiento en web, al final el objetivo se realizó con éxito aunque puede mejorar y en un futuro se implementará una versión final de este proyecto en una aplicación de escritorio para que mi padre pueda usar su herramienta con facilidad.