

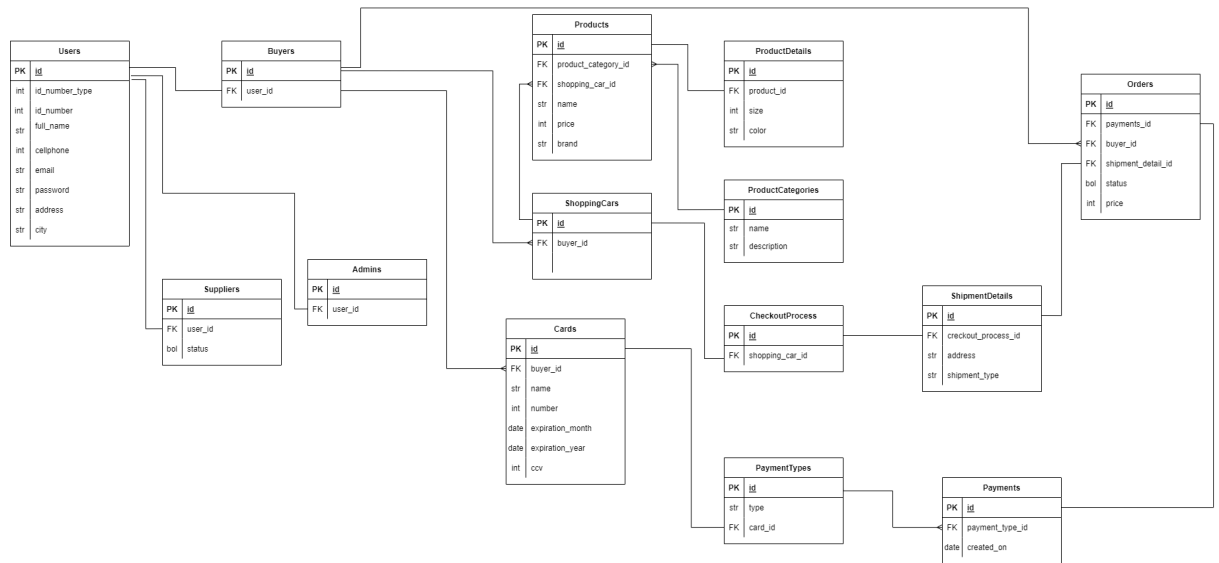
Bogotá, 23 de Febrero de 2022.

## Taller 1 - Bases de Datos 2

### Integrantes:

- Sergio Eduardo Peña Alméciga
- David Leonardo Lopez Forero
- Briannys Ahiram Paez Monserrate

### 1. Modelo relacional



### 2. Diccionario de datos

Nombre Archivo: Users			
Descripción: Archivo principal que contendrá información acerca de cada usuario			
Campo	Tipo	Tamaño	Descripción
id	clave primaria	1...1	indica el id consecutivo del usuario
id_number_type	numérico	1	indica el tipo de identificación del usuario
id_number	numérico	10	es el número de identificación del usuario
full_name	texto	50	es el nombre completo del usuario
cellphone	numérico	10	es el número celular del usuario
email	texto	20	es el email del usuario
password	texto	12	clave del usuario
address	texto	50	dirección de usuario
city	texto	10	ciudad de residencia del usuario
Nombre Archivo: Products			
Descripción: archivo dónde se contiene la información de los productos de los compradores			
Campo	Tipo	Tamaño	Descripción
id	clave primaria	1...1	indica el id consecutivo del producto
product_category_id	llave foránea	1...1	indica la categoría del producto
shopping_car_id	llave foránea	1...1	indica el carrito de compra del comprador
name	texto	50	es el nombre del producto
price	numérico	10	es el precio del producto
brand	texto	20	es la marca del producto
Nombre Archivo: Cards			
Descripción: archivo dónde se contiene la información de los productos de los compradores			
Campo	Tipo	Tamaño	Descripción
id	clave primaria	1...1	indica el id consecutivo de la tarjeta
buyer_id	llave foránea	1...1	indica la tarjeta del comprador
name	texto	50	es el nombre que se encuentra en la tarjeta
number	numérico	10	es el número de la tarjeta
expiration_month	date	2	es el mes de expiración de la tarjeta
expiration_year	date	4	es el año en el que expira la tarjeta
ccv	numérico	20	es el código de seguridad de la tarjeta
Nombre Archivo: ProductCategories			
Descripción: archivo dónde se contiene la información de la categoría del producto			
Campo	Tipo	Tamaño	Descripción
id	clave primaria	1...1	indica el id consecutivo de la categoría producto
name	texto	50	es el nombre de la categoría del producto
description	texto	50	es la descripción de la categoría del producto
Nombre Archivo: Orders			
Descripción: archivo dónde se contiene la información de la orden del comprador			
Campo	Tipo	Tamaño	Descripción
id	clave primaria	1...1	indica el id consecutivo de la orden
payments_id	llave foránea	1...1	indica el pago de la orden
buyer_id	llave foránea	1...1	indica el comprador al que se genera la orden
shipment_detail_id	llave foránea	1...1	indica el detalle del envío para realizar la orden
status	lógico	10	Activo / No Activo
price	numérico	10	el precio de toda la orden
Nombre Archivo: Payments			
Descripción: archivo dónde se contiene la información de la categoría del producto			
Campo	Tipo	Tamaño	Descripción
id	clave primaria	1...1	indica el id consecutivo del pago
payment_type_id	llave foránea	1...1	es el tipo de pago que se realizó
created_on	date	8	es la fecha de creación del pago

### 3. Vistas

```
CREATE VIEW marketplace.total_purchase_complete AS SELECT  
marketplace.buyer.id, marketplace.buyer.user_id, SUM(marketplace.order.price) AS  
total_purchase FROM marketplace.buyer INNER JOIN marketplace.order ON  
marketplace.buyer.id = marketplace.order.buyer_id WHERE marketplace.order.status  
= 'complete' GROUP BY marketplace.buyer.id;
```

Se crea una vista de la base de datos denominada “total\_purchase\_complete”. Seleccionamos los compradores para sumar el precio total de todas sus ordenes que tengan el status “completo”

```
CREATE VIEW marketplace.product_details_view AS SELECT  
marketplace.products.id, marketplace.products.name, marketplace.products.price,  
marketplace.products.brand, marketplace.product_categories.name AS category,  
marketplace.product_details.size, marketplace.product_details.color FROM  
marketplace.products INNER JOIN marketplace.product_details ON  
marketplace.products.id = marketplace.product_details.product_id INNER JOIN  
marketplace.product_categories ON marketplace.products.product_categories_id =  
marketplace.product_categories.id;
```

Se crea una vista llamada “product\_details\_view”. Seleccionamos de la tabla productos el nombre, el precio, la marca, la categoría, la talla, el color para así dar una vista al stock con todos los detalles.

- a. ¿Las vistas que decide crear a qué requerimiento no funcional obedecen? Seguridad o facilidad de consulta. ¿Deberían ser vistas materializadas? Argumente.

Estas vistas responden a lo uno y a lo otro ya que nos facilitan información importante para cierto tipo de usuario a la vez que restringen el acceso de dicha información a los usuarios restantes. No deberían ser materializadas ya que es información que es importante siempre consultar actualizada en tiempo real.

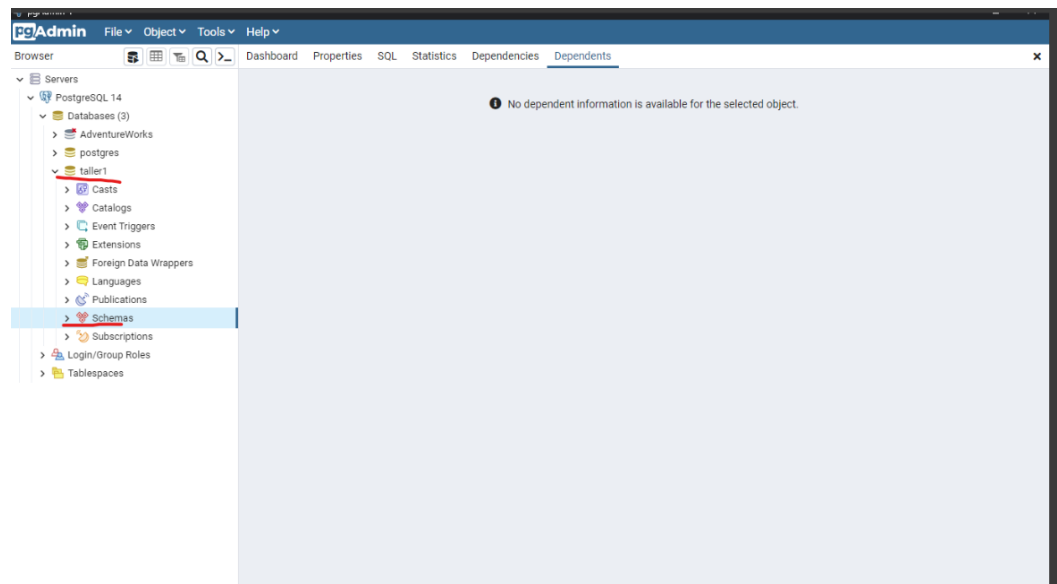
- b. ¿Cuáles consultas a la base de datos, a partir de los requerimientos dados, pueden optimizarse mediante índices? ¿De qué tipos deben ser dichos índices? Argumente.

Se optimizan las consultas para obtener el carrito de la compra que está relacionado con un proceso de compra, la tarjeta con la cual se realizó un pago y el id del pago que tiene una orden ¿, debido a que por medio de los índices podemos apuntar directamente a las ids que nos proveen esta información.

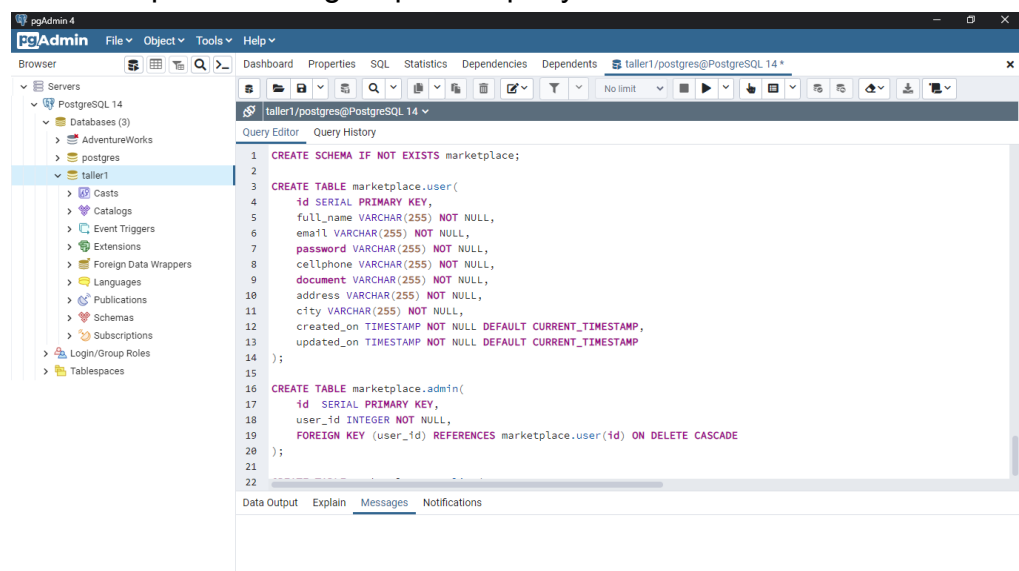
4. Script DDL a ser ejecutado en PostgreSQL. No se acepta el uso de herramientas de generación de código. Cree los índices previamente definidos utilizando los algoritmos dispuestos por el motor de base de datos

- [Scrip base de datos](#)

5. Evidencia de la ejecución del script en algún cliente de base de datos como pgAdmin.



- Se pone el codigo sql en el query tools



- Se corre la query y se visualiza como se crean tablas, vistas e indices.

