



UNIVERSIDAD DIEGO PORTALES

FACULTAD DE INGENIERÍA Y CIENCIAS

BASE DE DATOS AVANZADA

---

## Actividad 5

---

*Integrantes:*

Iván Cáceres  
Sebastián Cornejo  
Tobías Guerrero  
Marcelo Muñoz

27 de junio de 2022

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Entorno de Trabajo</b>	<b>3</b>
<b>3. Desarrollo</b>	<b>4</b>
3.1. Creación datos . . . . .	4
3.2. Parte 1 . . . . .	6
3.2.1. Creación Base de datos con csv . . . . .	6
3.2.2. Transacciones de Venta . . . . .	7
3.2.3. Consultas . . . . .	8
3.3. Parte 2 . . . . .	11
3.3.1. Nuevo Modelo . . . . .	11
3.3.2. Transacciones de Venta . . . . .	12
3.3.3. Consultas . . . . .	13
<b>4. Análisis</b>	<b>14</b>
<b>5. Conclusión</b>	<b>16</b>

## 1. Introducción

En la actividad 5 se trabaja con grandes volúmenes de datos y se intenta optimizar consultas típicas que se le pueden realizar al sistema. Para esta actividad se crean archivos .csv, uno para cada tabla, es decir, se crea cliente.csv, productos.csv y ventas.csv, con el fin de pasar estos archivos mediante un “COPY” a la base de datos, que es creada posteriormente a la creación de los .csv.

La actividad se divide en dos partes, la primera consta de la creación de la base de datos con las tres tablas, como se menciona anteriormente, luego se ejecutan diez transacciones de venta y se toma el tiempo correspondiente en cada una de ellas, para luego calcular un promedio, finalmente se realizan ciertas consultas que deben entregar determinadas respuestas. Para la parte 2, se busca optimizar todo lo hecho anteriormente, con el fin de optimizar los tiempos de consulta, es por esto que, se diseña un nuevo modelo de base de datos, una vez realizado esto, posteriormente se hace lo mismo que en la parte 1, es decir, se realizan diez transacciones de venta, con su tiempo y promedio, y finalmente se realizan las misma consultas de gestión.

## 2. Entorno de Trabajo

La actividad se realiza en un computador con las siguientes características (Software/Hardware):

Elemento	Valor
Nombre del SO	Microsoft Windows 10 Pro
Versión	10.0.19044 Compilación 19044
Descripción adicional del SO	No disponible
Fabricante del SO	Microsoft Corporation
Nombre del sistema	DESKTOP-BTF3JK5
Fabricante del sistema	HP
Modelo del sistema	HP EliteBook 830 G5
Tipo de sistema	x64-based PC
SKU del sistema	5FV92EC#ABM
Procesador	Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz, 1896 Mhz, 4 procesadores princi...
Versión y fecha de BIOS	HP Q78 Ver. 01.19.00, 13-01-2022
Versión de SMBIOS	3.1
Versión de controladora integr...	4.109
Modo de BIOS	UEFI
Fabricante de la placa base	HP
Producto placa base	83B3
Versión de la placa base	KBC Version 04.6D.00
Rol de plataforma	Móvil
Estado de arranque seguro	Activado
Configuración de PCR7	Se necesita elevación de privilegios para ver
Directorio de Windows	C:\WINDOWS
Directorio del sistema	C:\WINDOWS\system32
Dispositivo de arranque	\Device\HarddiskVolume1
Configuración regional	México
Capa de abstracción de hardw...	Versión = "10.0.19041.1566"
Nombre de usuario	DESKTOP-BTF3JK5\vaaaaaaaan
Zona horaria	Hora est. Sudamérica Pacifico
Memoria física instalada (RAM)	16,0 GB

Figura 1: Datos del sistema.

Elemento	Valor
Descripción	Unidad de disco
Fabricante	(Unidades de disco estándar)
Modelo	SAMSUNG MZVLB512HAJQ-000H1
Bytes/sector	512
Medio cargado	Sí
Tipo de medio	Fixed hard disk
Particiones	3
Bus SCSI	0
Unidades lógicas SCSI	0
Puerto SCSI	0
Id. de destino SCSI	0
Sectores/pista	63
Tamaño	476,94 GB (512.105.932.800 bytes)
Nº total de cilindros	62.260
Nº total de sectores	1.000.206.900
Nº total de pistas	15.876.300
Pistas/cilindro	255
Partición	Disco #0, partición #0
Tamaño de partición	260,00 MB (272.629.760 bytes)
Desplazamiento inicial de parti...	1.048.576 bytes
Partición	Disco #0, partición #1
Tamaño de partición	475,76 GB (510.847.352.832 bytes)
Desplazamiento inicial de parti...	290.455.552 bytes
Partición	Disco #0, partición #2
Tamaño de partición	922,00 MB (966.787.072 bytes)
Desplazamiento inicial de parti...	511.137.808.384 bytes

Figura 2: Especificaciones del disco.

Algo importante a mencionar, es que la unidad de almacenamiento que se utiliza es del tipo ssd M.2, por lo que los resultados esperados son de un tiempo reducido en comparación a una unidad hdd o ssd.

### 3. Desarrollo

Para cada parte de la actividad 5, se realizaron secciones para cada proceso, por cuestiones de orden y mejor apreciación del desarrollo de lo solicitado.

#### 3.1. Creación datos

En primer lugar, se debe preparar el ambiente de trabajo, es por esto que, mediante el lenguaje de programación c++ se realiza un código que ejecuta lo siguiente:

- Crea un archivo .csv para Clientes, que posee 5 millones de filas, en donde se utilizan parámetros como Codigo, Rut, Nombre, Direccion, estos parámetros son llenados con datos extraídos de PERSONAS1 (archivo .csv usado en entregas pasadas), y el delimitador entre estos parámetro es el carácter “|”.
- Crea un archivo .csv para Productos, que posee 1.000 productos diferentes, en donde se utilizan parámetros como Codigo, Nombre, Precio, estos parámetros son llenados de la siguiente manera: Codigo se completa a partir de los valores que toma i dentro de un for que empieza en 1 y llega hasta 1.000, para el nombre todos empiezan en p, pero tienen concatenado un i, que es el mismo i que codigo, finalmente para el precio se utiliza un random que va desde el 100 al 15.000, y el delimitador entre estos parámetros es el carácter “|”.
- Crea un archivo .csv para Ventas, que posee 200 millones de filas, en donde se utilizan parámetros como Codigo\_cliente, Codigo\_producto, Cantidad, Fecha\_venta, estos parámetros son llenados de la siguiente manera: Codigo\_cliente es extraído de clientes.csv, Codigo\_producto es extraído de productos.csv, Cantidad es generado por un random que va del 1 al 10 y Fecha\_venta es creado mediante 3 randoms, uno para día que va desde 1 a 28, uno para mes que va desde 1 a 12 y uno para año que va desde el 2020 al 2022, y el delimitador entre estos parámetros es el carácter “|”. Para este último archivo se consideraron alguna restricciones, las cuales son que a cada cliente se le haya vendido uniformemente un máximo de 40 productos diferentes, en distintas fechas y cantidades variables entre 1 y 10.

```

1  #include <iostream>
2  #include <fstream>
3  #include <ctime>
4  #include <cstdlib>
5  #include <string>
6  #include <sstream>
7  #include <vector>
8  #include <map>
9  using namespace std;
10
11 string int_to_string(int i)
12 {
13     stringstream ss;
14     ss << i;
15     return ss.str();
16 }
17
18 void productos(){
19     fstream fout; //puntero de fichero
20     remove("productos.csv");
21     fout.open("productos.csv", ios::out | ios::app); //crea el csv productos
22
23     srand((unsigned) time(0));
24     int randomNumber;
25
26     for(int i=1; i<=1000; i++){
27         randomNumber = (rand() % 15000) + 100;
28         fout << i << "|" << "p" << i << "|" << randomNumber << "\n";
29     }
30 }
31
32 void Clientes(){
33     fstream fin;
34     fin.open("../millones50_corregido.csv", ios::in);
35     remove("clientes.csv");
36     int number_of_lines = 5000000;
37     fstream fout;
38     fout.open("clientes.csv", ios::out | ios::app);

```

((a)) Parte 1

```

77
78 vector<string> codes_productos;
79
80 int number_of_products = 1000;
81 int number_of_clients = 5000000;
82
83 stringstream str(line);
84 for(int i=0; i<number_of_products; i++){
85     getline(fin2, line);
86     stringstream stream(line);
87
88     string codigo_producto, a;
89
90     getline(stream, codigo_producto, delimitador);
91     getline(stream, a, delimitador);
92     getline(stream, a, delimitador);
93
94     codes_productos.push_back(codigo_producto);
95 }
96
97 vector<string> codes_clientes;
98
99
100
101 for(int i=0; i<number_of_clients; i++){
102     getline(fin1, line);
103     stringstream stream(line);
104
105     string codigo_cliente, a;
106
107     getline(stream, codigo_cliente, delimitador);
108     getline(stream, a, delimitador);
109     getline(stream, a, delimitador);
110     getline(stream, a, delimitador);
111
112     codes_clientes.push_back(codigo_cliente);
113 }
114

```

((b)) Parte 3

```

39
40 string line;
41 char delimitador = '|';
42 for(int i=0; i<number_of_lines; i++){
43     getline(fin, line);
44     stringstream stream(line);
45     string Codigo, Rut, Nombre, Direccion, edad;
46
47     getline(stream, Codigo, delimitador);
48     Rut = Codigo;
49     getline(stream, Nombre, delimitador);
50     getline(stream, edad, delimitador);
51     getline(stream, Direccion, delimitador);
52
53     fout << Codigo << '|' << Rut << '|' << Nombre << '|' << Direccion << "\n";
54
55 }
56
57 cout << "Se ha creado clientes.csv con " << number_of_lines << " filas.";
58 }
59
60 void Ventas(){
61     fstream fout; //puntero de fichero
62     remove("ventas.csv");
63     fout.open("ventas.csv", ios::out | ios::app); //crea el csv productos
64
65     srand((unsigned) time(0));
66
67     int year, month, day;
68     int number_of_data = 200000000;
69
70     fstream fin1;
71     fin1.open("clientes.csv", ios::in);
72     fstream fin2;
73     fin2.open("productos.csv", ios::in);
74
75     char delimitador = '|';
76     string line;

```

((c)) Parte 2

```

115
116 //cout << "tamanno : " << codes_clientes.size();
117 //cout << codes_clientes[0];
118
119 vector<string> cantidad;
120 vector<string> fecha_venta;
121
122 //200 000 000 filas.
123 //number_of_clients
124
125
126 for(int j=0; j<number_of_clients; j++){
127     for(int i=0; i<40; i++){
128         int codigo_producto_temp = (rand() % 1000) + 1;
129         int cant_producto = (rand() % 10) + 1;
130
131         year = (rand() % (2022 - 2020)) + 2020;
132         month = (rand() % 12) + 1;
133         day = (rand() % 28) + 1;
134
135         string fecha_venta_f = int_to_string(year) + '-' +
136             int_to_string(month) + '-' + int_to_string(day);
137
138         fout << codes_clientes[j] << '|' << codigo_producto_temp
139             << '|' << cant_producto << '|' << fecha_venta_f << "\n";
140     }
141 }
142
143
144 void crear_archivos(){
145     productos();
146     Clientes();
147     Ventas();
148 }
149
150 int main(){
151     crear_archivos();
152
153     return 0;
154 }
155

```

((d)) Parte 4

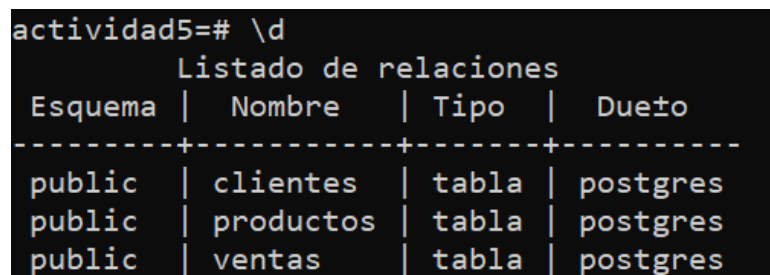
Figura 3: Código Creación Datos

### 3.2. Parte 1

Esta parte consta de la creación de una base de datos que sea óptima para realizar transacciones, de la ejecución de transacciones de ventas y de la respuesta a determinadas consultas.

#### 3.2.1. Creación Base de datos con csv

Una vez creados los datos para cada tabla, se diseña la base de datos que permite los mejores tiempos de respuesta para las típicas transacciones de ventas, y posee la siguiente estructura:



Listado de relaciones			
Esquema	Nombre	Tipo	Dueño
public	clientes	tabla	postgres
public	productos	tabla	postgres
public	ventas	tabla	postgres

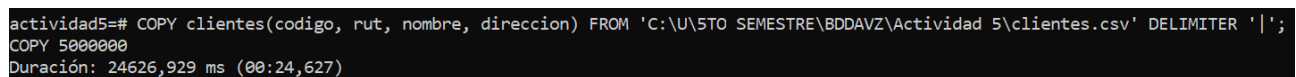
Figura 4: Estructura Base de datos 1

Para el diseño de esta base de datos lo óptimo es crear cada tabla con sus respectivos parámetros y sin índices ni llaves primarias, ya que para realizar transacciones de ventas, lo anterior es innecesario.

Ahora en cuanto a las tablas, a cada una se le realiza el siguiente comando, que es escrito de manera genérica:

```
1 COPY nombre_tabla(par metros) FROM 'Ubicacion archivo .csv correspondiente'
   DELIMITER '|';
```

Para cada COPY se toma el tiempo de ejecución correspondiente, ya que como hay un archivo de 1.000, otro de 5 millones y otro de 200 millones, lo más sensato es que esta última tome más tiempo que las anteriores, a continuación se adjuntan estos procesos.



```
actividad5=# COPY clientes(codigo, rut, nombre, direccion) FROM 'C:\U\5TO SEMESTRE\BDDAVZ\Actividad 5\clientes.csv' DELIMITER '|';
COPY 5000000
Duración: 24626,929 ms (00:24,627)
```

Figura 5: COPY Cliente con su tiempo de ejecución

```

actividad5=# COPY productos (codigo, nombre, precio) FROM 'C:\U\STO SEMESTRE\BDDAVZ\Actividad 5\productos.csv' DELIMITER '|';
COPY 1000
Duración: 16,272 ms

```

Figura 6: COPY Productos con su tiempo de ejecución

```

actividad5=# COPY ventas (codigo_cliente, codigo_producto, cantidad, fecha_venta) FROM 'C:\U\STO SEMESTRE\BDDAVZ\Actividad 5\ventas.csv' DELIMITER '|';
COPY 200000000
Duración: 727407,351 ms (12:07,407)

```

Figura 7: COPY Ventas con su tiempo de ejecución

COPY	Cientes	Productos	Ventas
Tiempo (ms)	24626,929	16,272	727407,351

Cuadro 1: Comparación de tiempos de cada COPY

### 3.2.2. Transacciones de Venta

El fin de la base de datos creada recientemente, es que sea óptima en cuanto a los tiempos de ejecución de transacciones de ventas, por ende a continuación se realizan 10 transacciones de venta del tipo “INSERT”.

```

actividad5=# insert into ventas values (4542619135, 1, 1, '2022-09-26'), (4542619135, 100, 1, '2022-09-26');
INSERT 0 2
Duración: 11,869 ms
actividad5=# insert into ventas values (1543550363, 23, 2, '2022-09-14');
INSERT 0 1
Duración: 1,282 ms
actividad5=# insert into ventas values (2245270413, 14, 3, '2022-10-17');
INSERT 0 1
Duración: 0,699 ms
actividad5=# insert into ventas values (652171354, 83, 4, '2022-10-17');
INSERT 0 1
Duración: 0,768 ms
actividad5=# insert into ventas values (233943311, 981, 5, '2022-10-24'), (233943311, 982, 3, '2022-10-24');
INSERT 0 2
Duración: 2,736 ms
actividad5=# insert into ventas values (3857413490, 838, 6, '2022-06-13');
INSERT 0 1
Duración: 0,638 ms
actividad5=# insert into ventas values (6385838394, 536, 7, '2022-04-27');
INSERT 0 1
Duración: 0,923 ms
actividad5=# insert into ventas values (4687586073, 182, 8, '2022-07-20');
INSERT 0 1
Duración: 0,348 ms
actividad5=# insert into ventas values (4687586073, 3, 9, '2022-02-13');
INSERT 0 1
Duración: 0,232 ms
actividad5=# insert into ventas values (4687586073, 1000, 10, '2022-03-22');
INSERT 0 1
Duración: 0,459 ms

```

Figura 8: Diez transacciones de venta con sus respectivos tiempos

Se mide el tiempo de ejecución de cada transacción, para finalmente calcular un promedio y corroborar si efectivamente lo mencionado antes se cumple. La tabla con los tiempos y el promedio se encuentra a continuación.

	Tiempo (ms)
Transacción 1	11,869
Transacción 2	1,282
Transacción 3	0,699
Transacción 4	0,768
Transacción 5	2,736
Transacción 6	0,638
Transacción 7	0,923
Transacción 8	0,348
Transacción 9	0,232
Transacción 10	0,459
<b>Promedio (ms)</b>	<b>1,9954</b>

Cuadro 2: Tiempo de cada transacción y el promedio de estas

### 3.2.3. Consultas

Una vez realizadas las transacciones, se procede a responder 3 consultas, las cuales se detallan a continuación. De estas consultas se analiza el comportamiento de los tiempos de ejecución.

La primera consulta solicita lo siguiente: Nombre de los clientes que han generado el mayor ingreso acumulado por las ventas que se les ha hecho.

Esta consulta se realiza con la siguiente sintaxis:

```
SELECT tabla.n, max FROM (SELECT c.nombre as n, sum(p.precio * v.cantidad) AS s
FROM clientes AS c, productos AS p, ventas AS v WHERE v.codigo_cliente = c.
codigo and v.codigo_producto = p.codigo group by 1) AS tabla ,(SELECT max(s)
FROM (SELECT c.nombre AS n, sum(p.precio * v.cantidad) AS s from clientes AS
c, productos AS p, ventas AS v WHERE v.codigo_cliente = c.codigo and v.
codigo_producto = p.codigo GROUP BY 1) AS tabla) AS max WHERE tabla.s=max;
```

A continuación se adjunta una imagen de la consulta, con su respuesta y su respectivo tiempo de ejecución.

```
Actividad=# select tabla.n, max from (select c.nombre as n, sum(p.precio * v.cantidad) as s from clientes as c, productos as p, ventas as v where v.codigo_cliente = c.codigo and v.codigo_producto = p.codigo group by 1) as tabla,(select max(s) from (select c.nombre as n, sum(p.precio * v.cantidad) as s from clientes as c, productos as p, ventas as v where v.codigo_cliente = c.codigo and v.codigo_producto = p.codigo group by 1) as tabla) as max where tabla.s=max;
n | max
--+--
BRPVTHLKNKJWCHCTQTRLHOENOLQVDPSCG3PBNJHAYEVLNI | 402237
RPQPBCHLHAUQZQBKXORBMVBPVCPVFBLEFETCRUNFDFQIYGH | 402237
(2 filas)
Duración: 1954350.419 ms (32:34,350)
```

Figura 9: Consulta 1

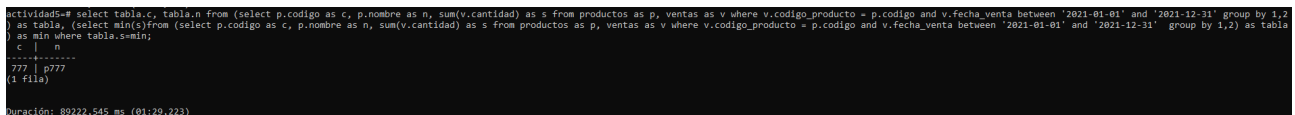


La segunda consulta solicita lo siguiente: Código y Nombre del producto menos vendido en un lapso de tiempo, en este caso a lo largo del año 2021.

Esta consulta se realiza con la siguiente sintáxis:

```
SELECT tabla.c, tabla.n FROM (SELECT p.codigo AS c, p.nombre AS n, sum(v.
    cantidad) AS s FROM productos AS p, ventas AS v WHERE v.codigo_producto = p.
    codigo and v.fecha_venta BETWEEN '2022-01-01' and '2022-12-31' GROUP BY 1,2)
AS tabla, (select min(s) FROM (SELECT p.codigo AS c, p.nombre AS n, sum(v.
    cantidad) AS s FROM productos AS p, ventas AS v WHERE v.codigo_producto = p.
    codigo and v.fecha_venta BETWEEN '2022-01-01' and '2022-12-31' GROUP BY 1,2)
AS tabla) AS min WHERE tabla.s=min;
```

A continuación se adjunta una imagen de la consulta, con su respuesta y su respectivo tiempo de ejecución.



```
actividad=# select tabla.c, tabla.n from (select p.codigo as c, p.nombre as n, sum(v.cantidad) as s from productos as p, ventas as v where v.codigo_producto = p.codigo and v.fecha_venta between '2021-01-01' and '2021-12-31' group by 1,2) as tabla, (select min(s) from (select p.codigo as c, p.nombre as n, sum(v.cantidad) as s from productos as p, ventas as v where v.codigo_producto = p.codigo and v.fecha_venta between '2021-01-01' and '2021-12-31' group by 1,2) as tabla) as min where tabla.s=min;
 c | n
---+---
777 | p777
(1 fila)

Duración: 89222,545 ms (01:29,223)
```

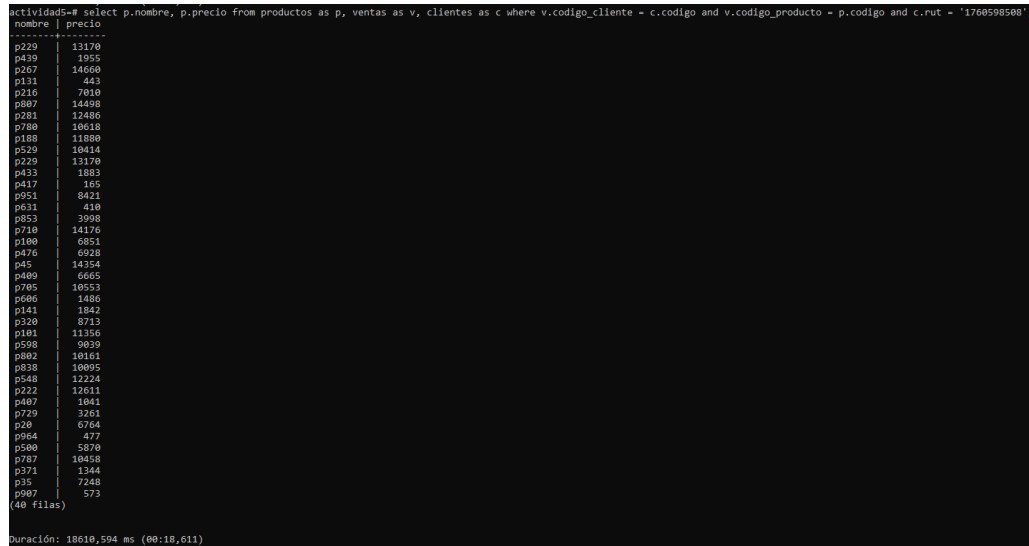
Figura 10: Consulta 2

La tercera consulta solicita lo siguiente: Nombre y Precio de los productos comprados por un cliente dado su rut.

Esta consulta se realiza con la siguiente sintáxis:

```
SELECT p.nombre, p.precio FROM productos AS p, ventas AS v, clientes AS c WHERE
    v.codigo_cliente = c.codigo and v.codigo_producto = p.codigo and c.rut =
    '1760598508';
```

A continuación se adjunta una imagen de la consulta, con su respuesta y su respectivo tiempo de ejecución.



```

actividad5=# select p.nombre, p.precio from productos as p, ventas as v, clientes as c where v.codigo_cliente = c.codigo and v.codigo_producto = p.codigo and c.rut = '1760598580';
 nombre | precio 
-----+-----
 p229   | 13170
 p439   | 1955
 p267   | 14660
 p131   | 443
 p216   | 7010
 p807   | 14490
 p281   | 12486
 p780   | 10618
 p188   | 11880
 p529   | 10414
 p229   | 13170
 p433   | 1883
 p417   | 165
 p951   | 8421
 p631   | 410
 p853   | 3998
 p710   | 14176
 p100   | 6851
 p476   | 6928
 p45    | 14354
 p409   | 6665
 p785   | 10553
 p686   | 1486
 p141   | 1842
 p320   | 8713
 p181   | 11356
 p598   | 9030
 p882   | 10161
 p838   | 10095
 p548   | 12224
 p222   | 11611
 p407   | 1041
 p729   | 3261
 p28    | 6764
 p964   | 477
 p500   | 5870
 p787   | 10458
 p371   | 1344
 p35    | 7248
 p907   | 573
(40 filas)

Duración: 18610,594 ms (00:18,611)

```

Figura 11: Consulta 3

A continuación se adjunta una tabla con los tiempos de ejecución de cada consulta.

	Consulta 1	Consulta 2	Consulta 3
Tiempo (ms)	1954350,419	89222,545	18610,594

Cuadro 3: Resumen de tiempos de cada consulta

### 3.3. Parte 2

Esta parte consta de la modificación de la base de datos de manera que sea óptima para realizar las consultas vistas anteriormente.

#### 3.3.1. Nuevo Modelo

Nuevamente se utilizan los datos que fueron creados al comienzo, se diseña la nueva base de datos que permite los mejores tiempos de respuesta para las 3 consultas que posteriormente se ven, esta base de datos lleva la siguiente estructura:

actividad5=# \d

Listado de relaciones

Esquema	Nombre	Tipo	Dueño
public	clientes	tabla	postgres
public	productos	tabla	postgres
public	resumenclientes	tabla	postgres
public	resumenventa2021	tabla	postgres
public	resumenventa2022	tabla	postgres
public	ventas	tabla	postgres

(6 filas)

Figura 12: Estructura base de datos 2

De la imagen se puede observar que se crean 3 tablas nuevas, las cuales se muestran a continuación. Cada una corresponde a información que es solicitada en las consultas originales, de esta forma se espera que al estar precalculada la información la consulta sea mucho más rápida.

```
actividad5=# create table resumenClientes as (select c.nombre as nombre, sum(p.precio * v.cantidad) as monto from clientes as c, productos as p, ventas as v where v.codigo_cliente = c.codigo and v.codigo_producto = p.codigo group by 1);
SELECT 5000000
Duración: 957582,660 ms (15:57,583)
```

Figura 13: Creación tabla Resumen Ingreso Clientes

```
actividad5=# create table resumenVenta2021 as (select p.codigo as c, p.nombre as n, sum(v.cantidad) as s from productos as p, ventas as v where v.codigo_producto = p.codigo and v.fecha_venta between '2021-01-01' and '2021-12-31' group by 1,2);
SELECT 1880
Duración: 43959,715 ms (00:43,960)
```

Figura 14: Creación tabla Resumen Ventas 2021

```
actividad5=# create table resumenVenta2022 as (select p.codigo as c, p.nombre as n, sum(v.cantidad) as s from productos as p, ventas as v where v.codigo_producto = p.codigo and v.fecha_venta between '2022-01-01' and '2022-12-31' group by 1,2);
SELECT 13
Duración: 9613,436 ms (00:09,613)
```

Figura 15: Creación tabla Resumen Ventas 2022

Posterior a la creación de las nuevas tablas, se definen 3 índices que serán un factor clave para reducir los tiempos para las consultas. El primer índice esta presente en la tabla Clientes y consta de un índice de dos parámetros, los cuales son Rut y Nombre, el segundo índice esta presente en la tabla Ventas y consta de un índice de dos parámetros, los cuales son Codigo\_cliente, Codigo\_producto, y el último índice esta presente en la tabla Productos y consta de un índice de dos parámetros, los cuales son Codigo y Nombre. A continuación se adjuntan imágenes de cada índice.

```
actividad5=# create index cliente on clientes (rut);  
CREATE INDEX  
Duración: 5383,519 ms (00:05,384)
```

Figura 16: Creación índice en tabla Cliente

```
actividad5=# create index venta on ventas (codigo_cliente, codigo_producto);  
CREATE INDEX  
Duración: 313494,584 ms (05:13,495)
```

Figura 17: Creación índice en tabla Ventas

```
actividad5=# create index producto on productos (codigo, nombre);  
CREATE INDEX
```

Figura 18: Creación índice en tabla Productos

### 3.3.2. Transacciones de Venta

Una vez terminada la estructura de la nueva base de datos, se procede a realizar las mismas 10 transacciones de venta que se desarrollaron en la parte 1.

Se mide el tiempo de ejecución de cada transacción, para finalmente calcular un promedio, esto sirve para ver si el tiempo de la ejecución de la transacción se ve afectado por el cambio de la estructura de la base de datos.

IMAGEN DE LAS TRANSACCIONES Y TABLA CON LOS TIEMPOS

---

### 3.3.3. Consultas

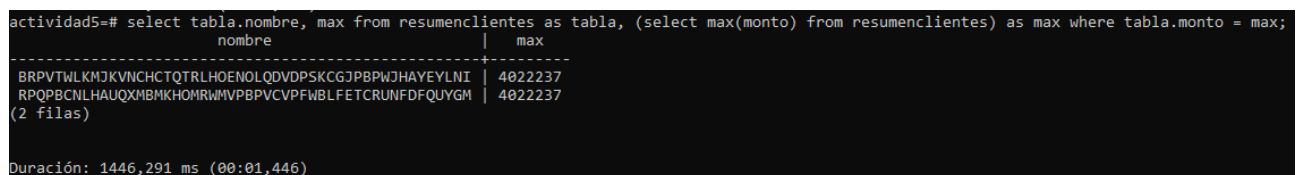
Una vez realizadas las transacciones, se procede a responden 3 consultas, las cuales se detallan a continuación. De estas consultas se analiza el tiempo de ejecución ya que la estructura de la base de datos se ha cambiado, por ende se espera que los tiempos mejoren considerablemente.

La primera consulta solicita lo siguiente: Nombre de los clientes que han generado el mayor ingreso acumulado por las ventas que se les ha hecho.

Esta consulta se realiza con la siguiente sintaxis:

```
SELECT tabla.nombre, max FROM resumenclientes AS tabla, (select max(monto) FROM
    resumenclientes) AS max where tabla.monto = max;
```

A continuación se adjunta una imagen de la consulta, con su respuesta y su respectivo tiempo de ejecución.



```
actividad5=# select tabla.nombre, max from resumenclientes as tabla, (select max(monto) from resumenclientes) as max where tabla.monto = max;
      nombre      | max
-----+-----
BRPVTWLMJKVNCHCTQTRLHOENOLQDVDPKCGJPBPWJHAYEYLN | 4022237
RQPQBNCNLHAUQXMBMKHOMRWMPBPVCPFWBLFETCRUNFDFQUYGM | 4022237
(2 filas)

Duración: 1446,291 ms (00:01,446)
```

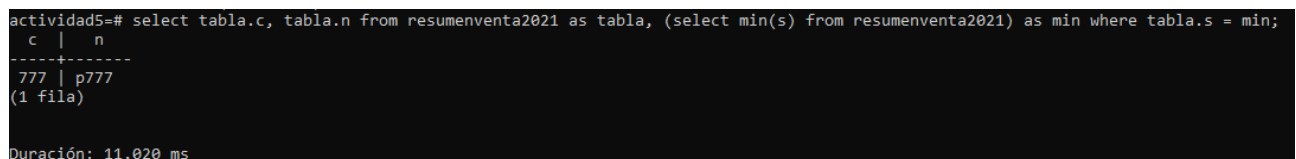
Figura 19: Consulta 1

La segunda consulta solicita lo siguiente: Código y Nombre del producto menos vendido en un lapso de tiempo.

Esta consulta se realiza con la siguiente sintaxis:

```
SELECT tabla.c, tabla.n FROM resumenventa2021 AS tabla, (select min(s) FROM
    resumenventa2021) AS min WHERE tabla.s = min;
```

A continuación se adjunta una imagen de la consulta, con su respuesta y su respectivo tiempo de ejecución.



```
actividad5=# select tabla.c, tabla.n from resumenventa2021 as tabla, (select min(s) from resumenventa2021) as min where tabla.s = min;
   c   | n
-----+-----
777 | p777
(1 fila)

Duración: 11,020 ms
```

Figura 20: Consulta 2

La tercera consulta solicita lo siguiente: Nombre y Precio de los productos comprados por un cliente dado su rut.

Esta consulta se realiza con la siguiente sintaxis:

```
SELECT p.nombre, p.precio FROM resumenprodclien AS p WHERE p.1 = '1760598508';
```

A continuación se adjunta una imagen de la consulta, con su respuesta y su respectivo tiempo de ejecución.

```

actividad5=# select p.nombre, p.precio from productos as p, ventas as v, clientes as c where v.codigo_cliente = c.codigo and v.codigo_producto = p.codigo and c.rut = '1768598508';
 nombre | precio
-----+-----
 p20    | 6764
 p35    | 7248
 p45    | 14354
 p100   | 6851
 p101   | 11356
 p131   | 443
 p141   | 1842
 p188   | 11880
 p216   | 7810
 p222   | 12611
 p229   | 13170
 p229   | 13170
 p267   | 14660
 p281   | 12486
 p320   | 8713
 p371   | 1344
 p407   | 1941
 p409   | 6665
 p417   | 165
 p433   | 1883
 p439   | 1955
 p476   | 6928
 p500   | 5870
 p529   | 10414
 p548   | 12224
 p598   | 9039
 p606   | 1486
 p631   | 410
 p705   | 10553
 p710   | 14176
 p729   | 3261
 p780   | 10618
 p787   | 10458
 p802   | 10161
 p807   | 14498
 p838   | 10095
 p853   | 3998
 p907   | 573
 p951   | 8421
 p964   | 477
(40 filas)

Duración: 0,880 ms

```

Figura 21: Consulta 3

A continuación se adjunta una tabla con los tiempos de ejecución de cada consulta.

	Consulta 1	Consulta 2	Consulta 3
Tiempo (ms)	1446,291	11,02	0,88

Cuadro 4: Resumen de tiempo de cada consulta

## 4. Análisis

De la parte 1 - cuadro 1, se comprueba lo que se menciona antes, que el tiempo depende netamente de la cantidad de datos del archivo y como ventas poseía 200 millones de datos, es el que tarda más.

De la parte 1 - cuadro 2, el promedio obtenido por las 10 transacciones de venta fue de 1,9954 ms, lo que significa que la base de datos fue diseñada de manera óptima para realizar transacciones de ventas, cumpliendo el objetivo de esa parte.

De la parte 1 - cuadro 3, se observa que los tiempos de la ejecución de las consultas son bastante diferentes en comparación a las transacciones realizadas anteriormente, esto es debido a que la base de datos fue creada para ser óptima para transacciones y no para este tipo de consultas, por lo que resulta normal que tarde mucho más tiempo en responder a cada una, ya que para dar respuesta, se debe consultar a más de una tabla, por lo que evidentemente este proceso consume

más tiempo, el ejemplo claro de esto, es la consulta 1 que tarda alrededor de 32 minutos en terminar de ejecutarse.

De la parte 2 - cuadro 4, se observa que los tiempos de la ejecución de las consultas redujeron bastante en comparación, esto es gracias a la nueva estructura que posee la base de , unas líneas más abajo se detalla el porque de esto.

A continuación se adjunta una tabla resumen con los tiempos de las 3 consultas de la parte 1 y de la parte 2, para poder realizar un análisis con mayor profundidad.

Tiempo ms	Consulta 1	Consulta 2	Consulta 3
Parte 1	1954350,419 ms	89222,545 ms	18610,594 ms
Parte 2	1146,291 ms	11,02 ms	0,88 ms

Cuadro 5: Tabla resumen Consultas Parte 1 y 2

De la parte 2, la consulta 1 y 2 tienen una gran mejora en el rendimiento, gracias a las tablas creadas, pensadas para cada consulta. La reducción del tiempo es debido a que como la operación más costosa es el join, estos datos se almacenan en las tablas resumen, de esta manera todos estos datos ya se encuentran precálculados y listos para su consulta, en cambio el modelo utilizado para la parte 1 requería de varios join, ya que se debía acceder a más tablas para dar las respuestas a las consultas. Ahora en cuanto a la consulta 3 igualmente existe una gran reducción en el tiempo de ejecución pero a diferencia de las otras 2 consultas no se optimizó mediante creación de tablas sino que se gracias a los índices creados la búsqueda fue mucho mas rápida.

Finalmente se adjunta un gráfico para apreciar visualmente la reducción de los tiempos de las consultas en cada parte.



Figura 22: Gráfico resumen de consultas de cada parte

Tiempo ms	Promedio Transacciones
Parte 1	1,9954 ms
Parte 2	5,9862 ms

Cuadro 6: Tabla resumen Transacciones Parte 1 y 2

Podemos ver que en el modelo de gestión al insertar a alguna tabla se deben actualizar las otras que obtienen datos de estas, observando que mínimo debería tomar 3 veces más el insertar ya que debe actualizar 3 tablas aparte de insertar la venta.

## 5. Conclusión

Es posible observar que en la parte 1 con la base de datos transaccional, las transacciones son mucho más rápidas que en el modelo de la parte 2, esto por que en la parte 2 es necesario actualizar todas las tablas creadas para las consultas de gestión.

También en la parte 1, las consultas demoran bastante tiempo, esto por el tipo de consulta con tantos joins entre tablas, mientras que en la parte dos apenas en unos segundos tiene todas estas consultas.

Algunos de los problemas del modelo de la parte 2, es por ejemplo que consume muchos recursos en temas de almacenamiento, también hay muchas tablas que mantener, que deben ser actualizadas para verificar una consistencia (respuestas iguales en consultas equivalentes), incluso



hay un problema de redundancia al haber tablas con datos repetidos, todo esto para una más rápida respuesta a consultas gestionales.

Finalmente se concluye que el modelo de gestión esta hecho para resolver consultas, mientras que el transaccional para simplemente hacer transacciones y tomando en cuenta lo visto en clase, la idea óptima es separar procesos transaccionales y de gestion por ejemplo en 2 bases de datos distintas, sincronizadas.