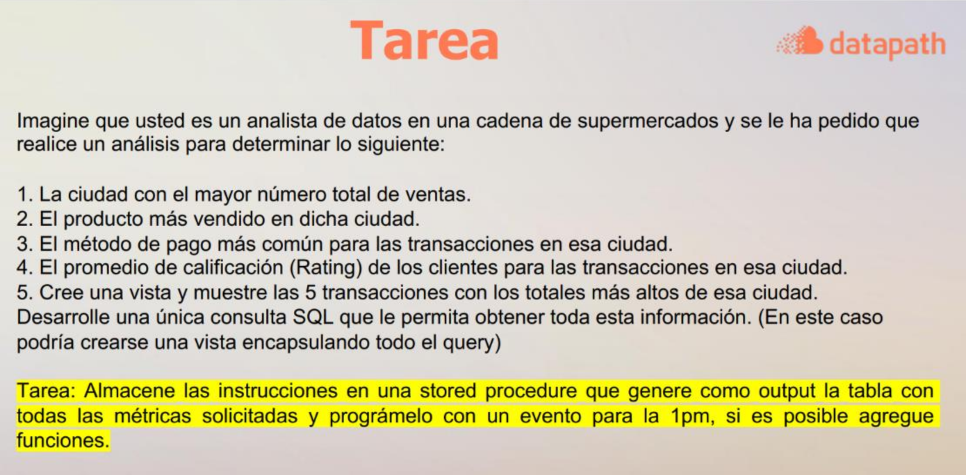
Tarea 2 Base de Datos Relacional

Nombre: Miguel Angel Tinoco Arroyo



Tarea

Imagine que usted es un analista de datos en una cadena de supermercados y se le ha pedido que realice un análisis para determinar lo siguiente:

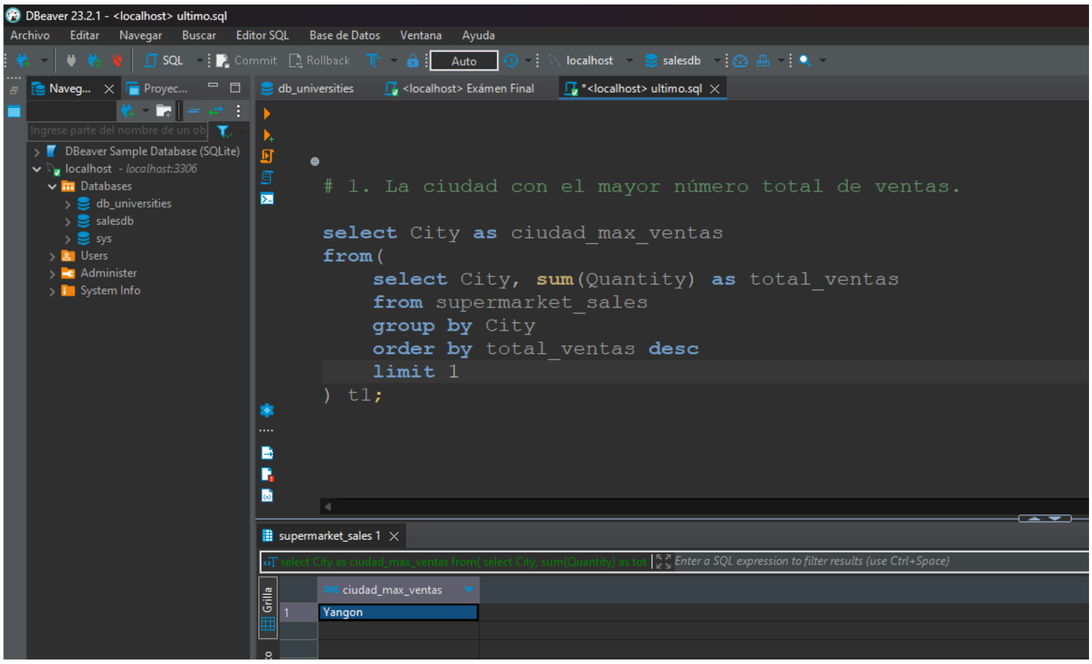
1. La ciudad con el mayor número total de ventas.
2. El producto más vendido en dicha ciudad.
3. El método de pago más común para las transacciones en esa ciudad.
4. El promedio de calificación (Rating) de los clientes para las transacciones en esa ciudad.
5. Cree una vista y muestre las 5 transacciones con los totales más altos de esa ciudad.

Desarrolle una única consulta SQL que le permita obtener toda esta información. (En este caso podría crearse una vista encapsulando todo el query)

Tarea: Almacene las instrucciones en una stored procedure que genere como output la tabla con todas las métricas solicitadas y prográmelo con un evento para la 1pm, si es posible agregue funciones.

Solución:

1.

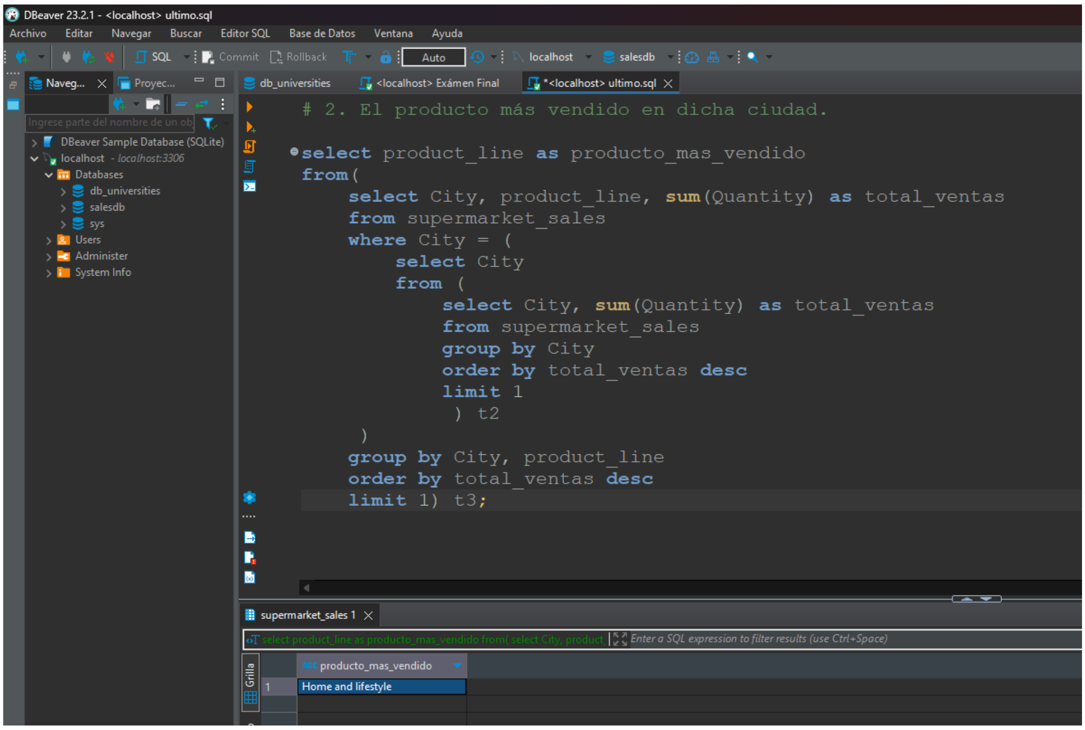


DBeaver 23.2.1 - <localhost> ultimo.sql
Archivo Editar Navegar Buscar Editor SQL Base de Datos Ventana Ayuda
SQL Commit Rollback Auto localhost salesdb
Naveg... Proyec...
DBeaver Sample Database (SQLite)
localhost - localhost:3306
Databases
db_universities
salesdb
sys
Users
Administer
System Info
db_universities
<localhost> Exámen Final
*<localhost> ultimo.sql
supermarket_sales 1
Enter a SQL expression to filter results (use Ctrl+Space)
Grilla

```sql
# 1. La ciudad con el mayor número total de ventas.

select City as ciudad_max_ventas
from(
    select City, sum(Quantity) as total_ventas
    from supermarket_sales
    group by City
    order by total_ventas desc
    limit 1
) t1;
```

	ciudad_max_ventas
1	Yangon

2.

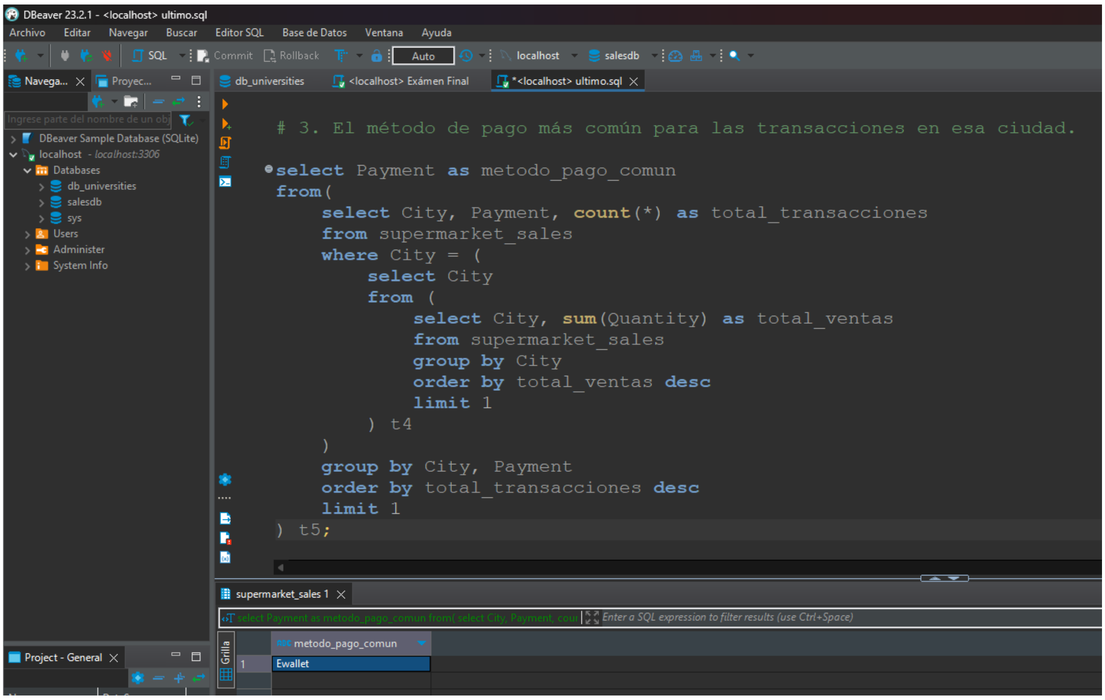


```sql
# 2. El producto más vendido en dicha ciudad.

select product_line as producto_mas_vendido
from(
    select City, product_line, sum(Quantity) as total_ventas
    from supermarket_sales
    where City = (
        select City
        from (
            select City, sum(Quantity) as total_ventas
            from supermarket_sales
            group by City
            order by total_ventas desc
            limit 1
             ) t2
         )
    group by City, product_line
    order by total_ventas desc
    limit 1) t3;
```

select product_line as producto_mas_vendido from(select City, product | Enter a SQL expression to filter results (use Ctrl+Space)

	producto_mas_vendido
1	Home and lifestyle

3.

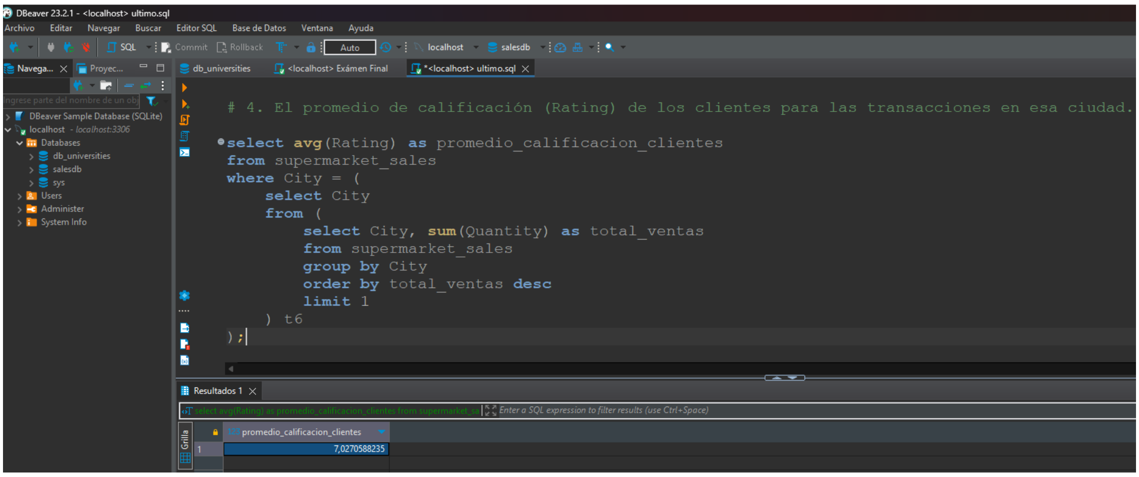


```sql
# 3. El método de pago más común para las transacciones en esa ciudad.

select Payment as metodo_pago_comun
from(
    select City, Payment, count(*) as total_transacciones
    from supermarket_sales
    where City = (
        select City
        from (
            select City, sum(Quantity) as total_ventas
            from supermarket_sales
            group by City
            order by total_ventas desc
            limit 1
        ) t4
    )
    group by City, Payment
    order by total_transacciones desc
    limit 1
) t5;
```

select Payment as metodo_pago_comun from(select City, Payment, cour | Enter a SQL expression to filter results (use Ctrl+Space)

	metodo_pago_comun
1	Ewallet

4.



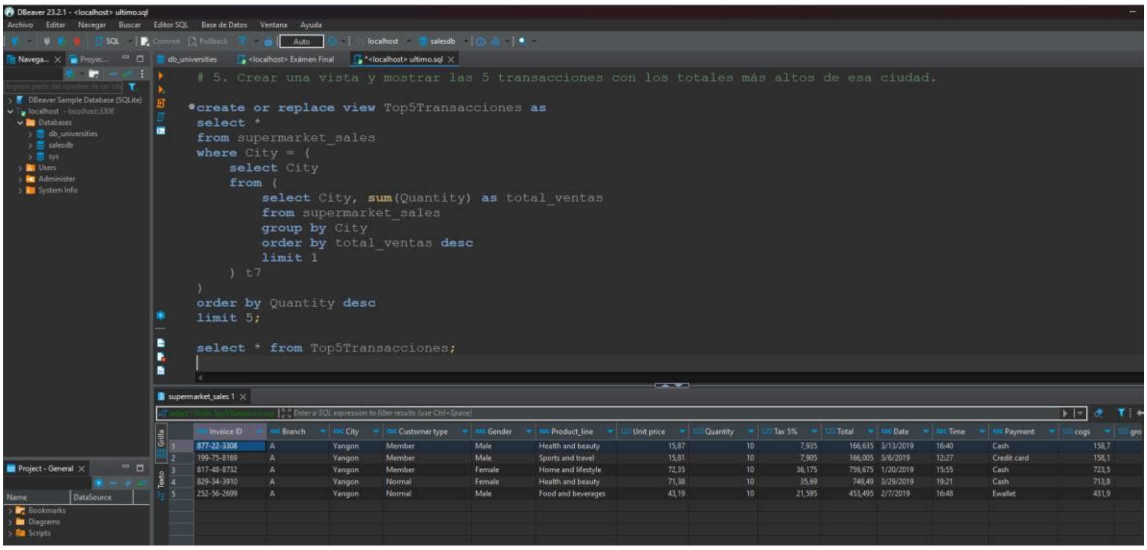
```
# 4. El promedio de calificación (Rating) de los clientes para las transacciones en esa ciudad.

select avg(Rating) as promedio_calificacion_clientes
from supermarket_sales
where City = (
    select City
    from (
        select City, sum(Quantity) as total_ventas
        from supermarket_sales
        group by City
        order by total_ventas desc
        limit 1
    ) t6
);
```

Resultados 1

select avg(Rating) as promedio_calificacion_clientes from supermarket_sa

	promedio_calificacion_clientes
1	7,0270588235

5.

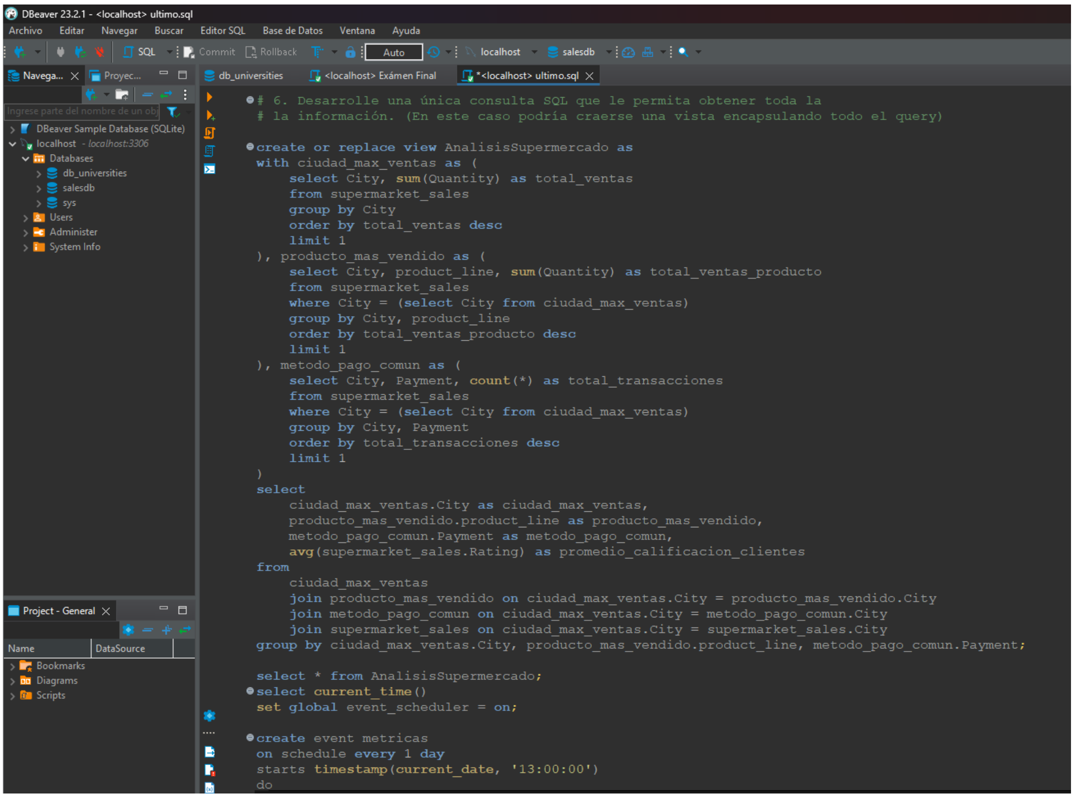


```
# 5. Crear una vista y mostrar las 5 transacciones con los totales más altos de esa ciudad.

create or replace view Top5Transacciones as
select *
from supermarket_sales
where City = (
    select City
    from (
        select City, sum(Quantity) as total_ventas
        from supermarket_sales
        group by City
        order by total_ventas desc
        limit 1
    ) t7
)
order by Quantity desc
limit 5;

select * from Top5Transacciones;
```

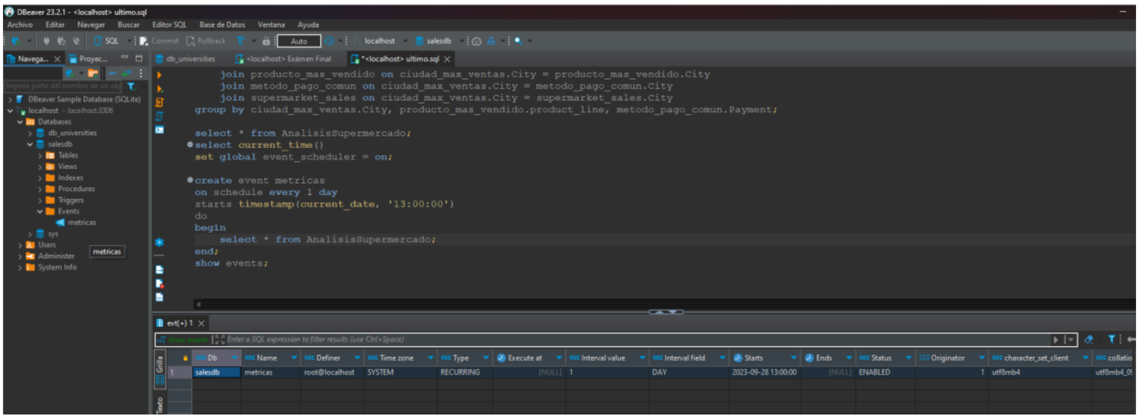
supermarket_sales 1

	Invoice ID	Branch	City	Customer type	Gender	Product_line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	cogs	gro
1	877-22-3308	A	Yangon	Member	Male	Health and beauty	15,87	10	7,935	166,635	3/13/2019	16:40	Cash	158,7	
2	199-75-8169	A	Yangon	Member	Male	Sports and travel	15,81	10	7,905	166,005	3/6/2019	12:27	Credit card	158,1	
3	817-48-8732	A	Yangon	Member	Female	Home and lifestyle	72,35	10	36,175	759,675	1/20/2019	15:55	Cash	723,5	
4	829-34-3910	A	Yangon	Normal	Female	Health and beauty	71,38	10	35,69	749,49	3/29/2019	19:21	Cash	713,8	
5	252-56-2699	A	Yangon	Normal	Male	Food and beverages	43,19	10	21,595	453,495	2/7/2019	16:48	Ewallet	431,9	

6.



```sql
# 6. Desarrolle una única consulta SQL que le permita obtener toda la
# la información. (En este caso podria craerse una vista encapsulando todo el query)

create or replace view AnalisisSupermercado as
with ciudad_max_ventas as (
    select City, sum(Quantity) as total_ventas
    from supermarket_sales
    group by City
    order by total_ventas desc
    limit 1
), producto_mas_vendido as (
    select City, product_line, sum(Quantity) as total_ventas_producto
    from supermarket_sales
    where City = (select City from ciudad_max_ventas)
    group by City, product_line
    order by total_ventas_producto desc
    limit 1
), metodo_pago_comun as (
    select City, Payment, count(*) as total_transacciones
    from supermarket_sales
    where City = (select City from ciudad_max_ventas)
    group by City, Payment
    order by total_transacciones desc
    limit 1
)
select
    ciudad_max_ventas.City as ciudad_max_ventas,
    producto_mas_vendido.product_line as producto_mas_vendido,
    metodo_pago_comun.Payment as metodo_pago_comun,
    avg(supermarket_sales.Rating) as promedio_calificacion_clientes
from
    ciudad_max_ventas
    join producto_mas_vendido on ciudad_max_ventas.City = producto_mas_vendido.City
    join metodo_pago_comun on ciudad_max_ventas.City = metodo_pago_comun.City
    join supermarket_sales on ciudad_max_ventas.City = supermarket_sales.City
group by ciudad_max_ventas.City, producto_mas_vendido.product_line, metodo_pago_comun.Payment;

select * from AnalisisSupermercado;
select current_time()
set global event_scheduler = on;

create event metricas
on schedule every 1 day
starts timestamp(current_date, '13:00:00')
do
```



```sql
    join producto_mas_vendido on ciudad_max_ventas.City = producto_mas_vendido.City
    join metodo_pago_comun on ciudad_max_ventas.City = metodo_pago_comun.City
    join supermarket_sales on ciudad_max_ventas.City = supermarket_sales.City
group by ciudad_max_ventas.City, producto_mas_vendido.product_line, metodo_pago_comun.Payment;

select * from AnalisisSupermercado;
select current_time()
set global event_scheduler = on;

create event metricas
on schedule every 1 day
starts timestamp(current_date, '13:00:00')
do
begin
    select * from AnalisisSupermercado;
end;
show events;
```

	Db	Name	Definer	Time zone	Type	Execute at	Interval value	Interval field	Starts	Ends	Status	Originator	character_set_client	collatio
1	salesdb	metricas	root@localhost	SYSTEM	RECURRING	[NULL]	1	DAY	2023-09-28 13:00:00	[NULL]	ENABLED	1	utf8mb4	utf8mb4_0