

ANALITICA

SANYER DUQUE HOYOS

MANIZALES - CALDAS

JUNIO 2025

**Casos de Uso:** La compañía ABC es una empresa de producción de rosas, dicha compañía cuenta con 15 hectáreas disponibles para siembra en la sabana de Bogotá. La compañía tiene distribuida sus siembras en Bloques, cada bloque cuenta con camas.

En cada cama se siembran rosales de diferentes colores, los cuales producen tallos de rosa cada 13 semanas. Al acabar el ciclo de floración los rosales se podan y las rosas obtenidas se envían al inventario de postcosecha donde son almacenadas y refrigeradas.

Después de cada poda el ciclo de floración inicia nuevamente.

### Punto 1:

Diseñe un diagrama Entidad-Relación en donde se pueda almacenar la información de cada ciclo de floración, debe quedar registrado la fecha en la que se realiza cada poda, el operario que realiza la labor y la ubicación exacta de cada una. Adicionalmente en otra tabla debe almacenar los tallos podados que serán enviados al inventario de postcosecha para ser manufacturados por la empresa.

## SOLUCIÓN

### Entidades y atributos:

tb_Ciclo_Facturacion		
<b>PK</b>	id_ciclo_facturacion	INT
	fecha_inicio	DATETIME
	fecha_fin	DATETIME

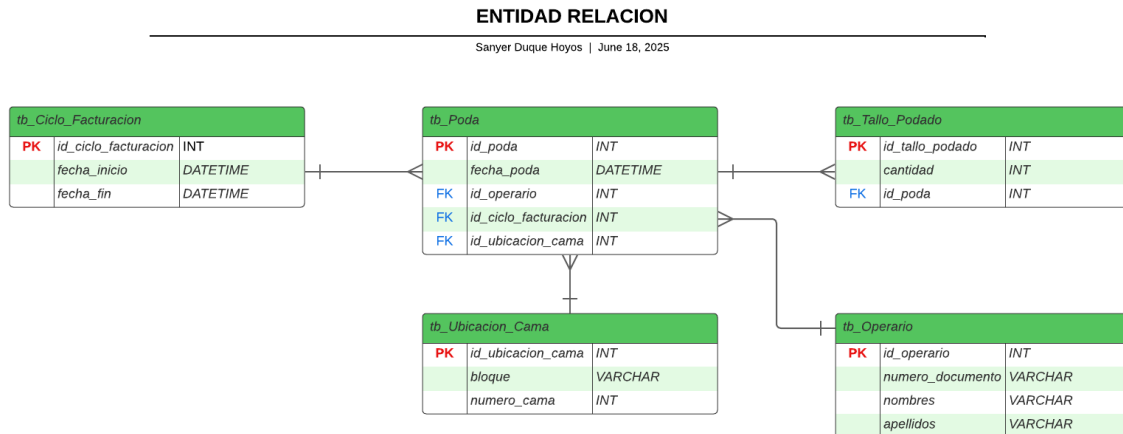
tb_Poda		
<b>PK</b>	id_poda	INT
	fecha_poda	DATETIME
<b>FK</b>	id_operario	INT
<b>FK</b>	id_ciclo_facturacion	INT
<b>FK</b>	id_ubicacion_cama	INT

tb_Ubicacion_Cama		
<b>PK</b>	id_ubicacion_cama	INT
	bloque	VARCHAR
	numero_cama	INT

tb_Operario		
<b>PK</b>	id_operario	INT
	numero_documento	VARCHAR
	nombres	VARCHAR
	apellidos	VARCHAR

tb_Tallo_Podado		
<b>PK</b>	id_tallo_podado	INT
	cantidad	INT
<b>FK</b>	id_poda	INT

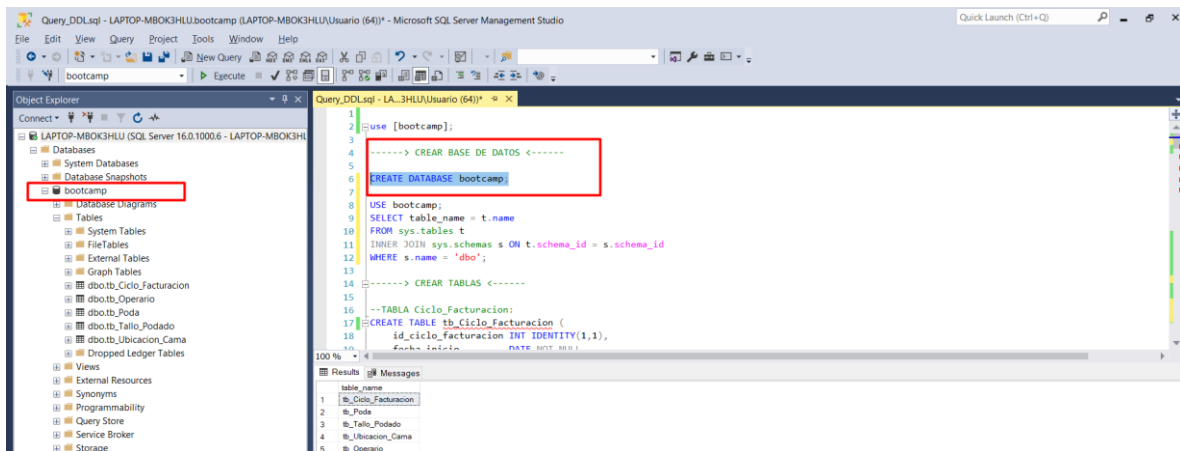
## Entidad-Relación (Lucidchart):



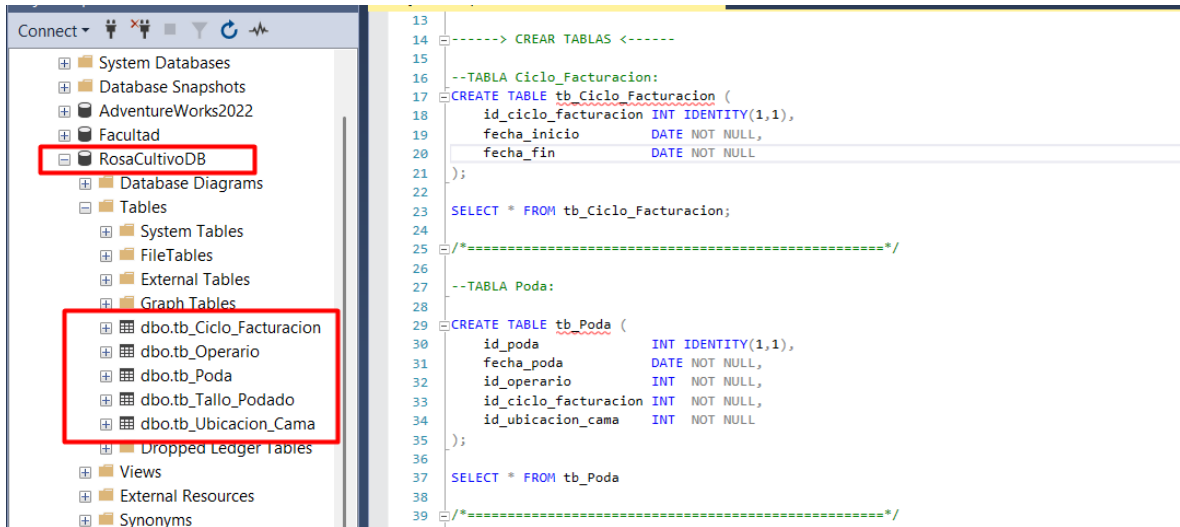
## Punto 2:

Implemente la base de datos diseñada en el punto 1, e ingrese datos de prueba. Para esta tarea puede instalar en su máquina SQL Server Express.

## -Creación de BD

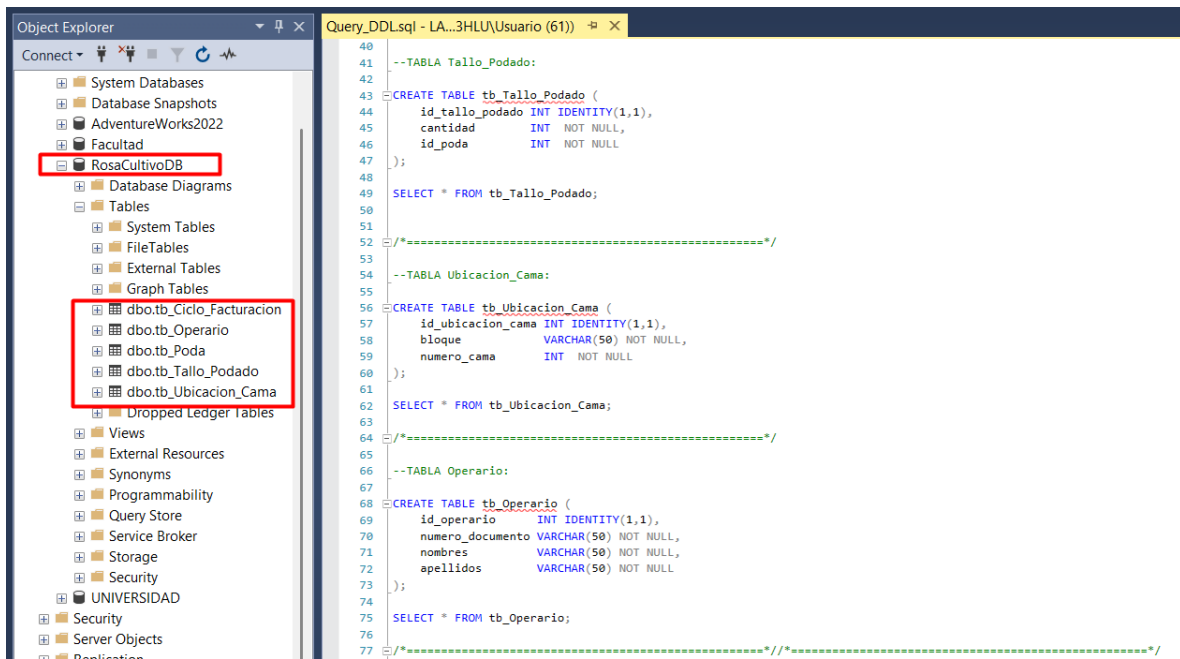


## -Creación de tablas:



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure for 'RosaCultivoDB'. The 'Tables' folder is expanded, showing the following tables: 'dbo.tb\_Ciclo\_Facturacion', 'dbo.tb\_Operario', 'dbo.tb\_Poda', 'dbo.tb\_Tallo\_Podado', and 'dbo.tb\_Ubicacion\_Cama'. These tables are highlighted with a red box. On the right, the Query Editor shows the T-SQL code for creating these tables. The code is as follows:

```
13
14 -----> CREAM TABLAS <-----
15
16 --TABLA Ciclo_Facturacion:
17 CREATE TABLE tb_Ciclo_Facturacion (
18     id_ciclo_facturacion INT IDENTITY(1,1),
19     fecha_inicio DATE NOT NULL,
20     fecha_fin DATE NOT NULL
21 );
22
23 SELECT * FROM tb_Ciclo_Facturacion;
24
25 /*=====*/
26
27 --TABLA Poda:
28
29 CREATE TABLE tb_Poda (
30     id_poda INT IDENTITY(1,1),
31     fecha_poda DATE NOT NULL,
32     id_operario INT NOT NULL,
33     id_ciclo_facturacion INT NOT NULL,
34     id_ubicacion_cama INT NOT NULL
35 );
36
37 SELECT * FROM tb_Poda
38
39 /*=====*/
```



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure for 'RosaCultivoDB'. The 'Tables' folder is expanded, showing the following tables: 'dbo.tb\_Ciclo\_Facturacion', 'dbo.tb\_Operario', 'dbo.tb\_Poda', 'dbo.tb\_Tallo\_Podado', and 'dbo.tb\_Ubicacion\_Cama'. These tables are highlighted with a red box. On the right, the Query Editor shows the T-SQL code for creating these tables. The code is as follows:

```
40
41 --TABLA Tallo_Podado:
42
43 CREATE TABLE tb_Tallo_Podado (
44     id_tallo_podado INT IDENTITY(1,1),
45     cantidad INT NOT NULL,
46     id_poda INT NOT NULL
47 );
48
49 SELECT * FROM tb_Tallo_Podado;
50
51 /*=====*/
52
53 --TABLA Ubicacion_Cama:
54
55 CREATE TABLE tb_Ubicacion_Cama (
56     id_ubicacion_cama INT IDENTITY(1,1),
57     bloque VARCHAR(50) NOT NULL,
58     numero_cama INT NOT NULL
59 );
60
61 SELECT * FROM tb_Ubicacion_Cama;
62
63 /*=====*/
64
65 --TABLA Operario:
66
67 CREATE TABLE tb_Operario (
68     id_operario INT IDENTITY(1,1),
69     numero_documento VARCHAR(50) NOT NULL,
70     nombres VARCHAR(50) NOT NULL,
71     apellidos VARCHAR(50) NOT NULL
72 );
73
74 SELECT * FROM tb_Operario;
75
76 /*=====*/
77
```

## -Crear llaves primarias:

The screenshot shows the SQL Server Enterprise Manager interface on the left and a query window on the right. In the Enterprise Manager, the 'Keys' folder under 'dbo.tb\_Ciclo\_Facturacion' is highlighted with a red box. The query window displays the following SQL code:

```
70 numero_documento VARCHAR(50) NOT NULL,  
71 nombres VARCHAR(50) NOT NULL,  
72 apellidos VARCHAR(50) NOT NULL  
73 );  
74  
75 SELECT * FROM tb_Operario;  
76  
77 /*-----*/  
78  
79 --LLAVES PRIMARIAS:  
80  
81 ALTER TABLE tb_Ciclo_Facturacion  
82 ADD CONSTRAINT PK_tb_Ciclo_Facturacion PRIMARY KEY (id_ciclo_facturacion);  
83  
84 ALTER TABLE tb_Poda  
85 ADD CONSTRAINT PK_tb_Poda PRIMARY KEY (id_poda);  
86  
87 ALTER TABLE tb_Tallo_Podado  
88 ADD CONSTRAINT PK_tb_Tallo_Podado PRIMARY KEY (id_tallo_podado);  
89  
90 ALTER TABLE tb_Ubicacion_Cama  
91 ADD CONSTRAINT PK_tb_Ubicacion_Cama PRIMARY KEY (id_ubicacion_cama);  
92  
93 ALTER TABLE tb_Operario  
94 ADD CONSTRAINT PK_tb_Operario PRIMARY KEY (id_operario);  
95
```

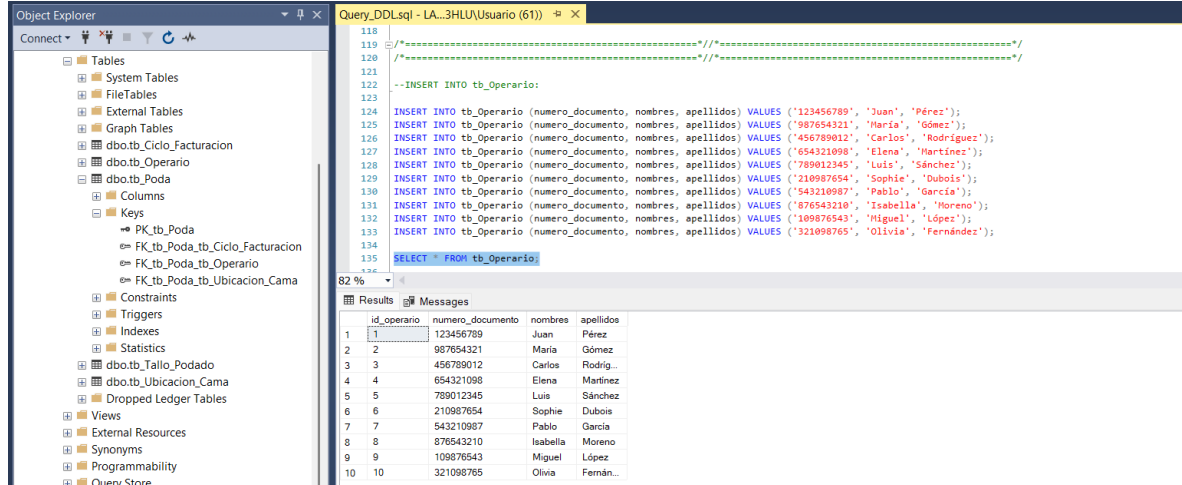
## -Crear llaves foráneas:

The screenshot shows the SQL Server Enterprise Manager interface on the left and a query window on the right. In the Enterprise Manager, the 'Keys' folder under 'dbo.tb\_Poda' is highlighted with a red box, showing three foreign keys: 'FK\_tb\_Poda\_tb\_Ciclo\_Facturacion', 'FK\_tb\_Poda\_tb\_Operario', and 'FK\_tb\_Poda\_tb\_Ubicacion\_Cama'. The query window displays the following SQL code:

```
94 ALTER TABLE tb_Operario  
95 ADD CONSTRAINT PK_tb_Operario PRIMARY KEY (id_operario);  
96  
97 --LLAVES FORANEAS:  
98  
99 ALTER TABLE tb_Tallo_Podado  
100 ADD CONSTRAINT FK_tb_Tallo_Podado_tb_Poda  
101 FOREIGN KEY (id_poda)  
102 REFERENCES tb_Poda(id_poda);  
103  
104 ALTER TABLE tb_Poda  
105 ADD CONSTRAINT FK_tb_Poda_tb_Operario  
106 FOREIGN KEY (id_operario)  
107 REFERENCES tb_Operario(id_operario);  
108  
109 ALTER TABLE tb_Poda  
110 ADD CONSTRAINT FK_tb_Poda_tb_Ciclo_Facturacion  
111 FOREIGN KEY (id_ciclo_facturacion)  
112 REFERENCES tb_Ciclo_Facturacion(id_ciclo_facturacion);  
113  
114 ALTER TABLE tb_Poda  
115 ADD CONSTRAINT FK_tb_Poda_tb_Ubicacion_Cama  
116 FOREIGN KEY (id_ubicacion_cama)  
117 REFERENCES tb_Ubicacion_Cama(id_ubicacion_cama);  
118
```

## -Insertar datos de prueba:

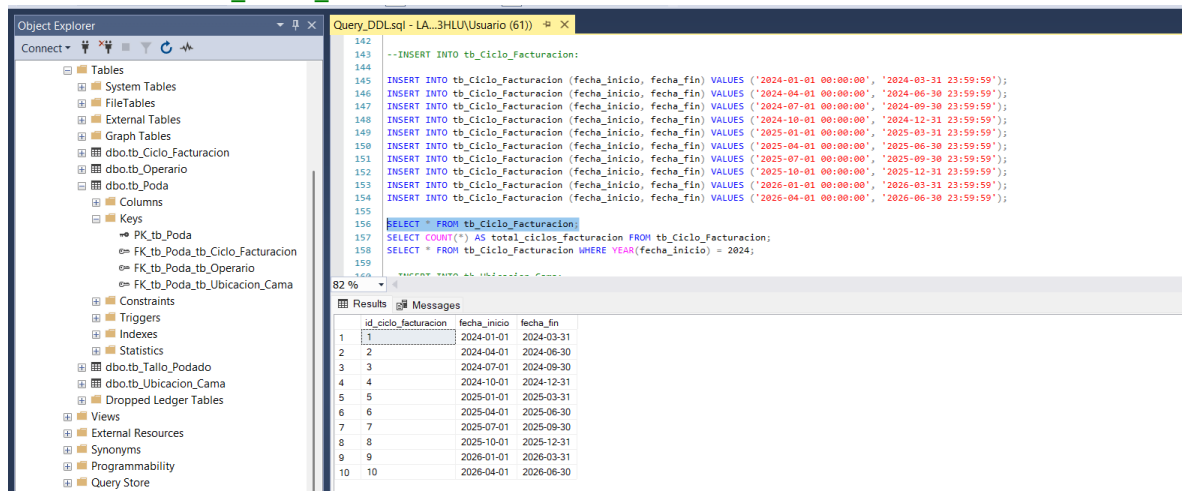
--INSERT INTO tb\_Operario:



```
119
120
121
122 --INSERT INTO tb_Operario:
123
124 INSERT INTO tb_Operario (numero_documento, nombres, apellidos) VALUES ('123456789', 'Juan', 'Pérez');
125 INSERT INTO tb_Operario (numero_documento, nombres, apellidos) VALUES ('987654321', 'María', 'Gómez');
126 INSERT INTO tb_Operario (numero_documento, nombres, apellidos) VALUES ('456789012', 'Carlos', 'Rodríguez');
127 INSERT INTO tb_Operario (numero_documento, nombres, apellidos) VALUES ('654321098', 'Elena', 'Martínez');
128 INSERT INTO tb_Operario (numero_documento, nombres, apellidos) VALUES ('789012345', 'Luis', 'Sánchez');
129 INSERT INTO tb_Operario (numero_documento, nombres, apellidos) VALUES ('210987654', 'Sophie', 'Dubois');
130 INSERT INTO tb_Operario (numero_documento, nombres, apellidos) VALUES ('543210987', 'Pablo', 'García');
131 INSERT INTO tb_Operario (numero_documento, nombres, apellidos) VALUES ('876543210', 'Isabella', 'Moreno');
132 INSERT INTO tb_Operario (numero_documento, nombres, apellidos) VALUES ('109876543', 'Miguel', 'López');
133 INSERT INTO tb_Operario (numero_documento, nombres, apellidos) VALUES ('321098765', 'Olivia', 'Fernández');
134
135 SELECT * FROM tb_Operario;
```

id_operario	numero_documento	nombres	apellidos
1	123456789	Juan	Pérez
2	987654321	María	Gómez
3	456789012	Carlos	Rodríguez
4	654321098	Elena	Martínez
5	789012345	Luis	Sánchez
6	210987654	Sophie	Dubois
7	543210987	Pablo	García
8	876543210	Isabella	Moreno
9	109876543	Miguel	López
10	321098765	Olivia	Fernández

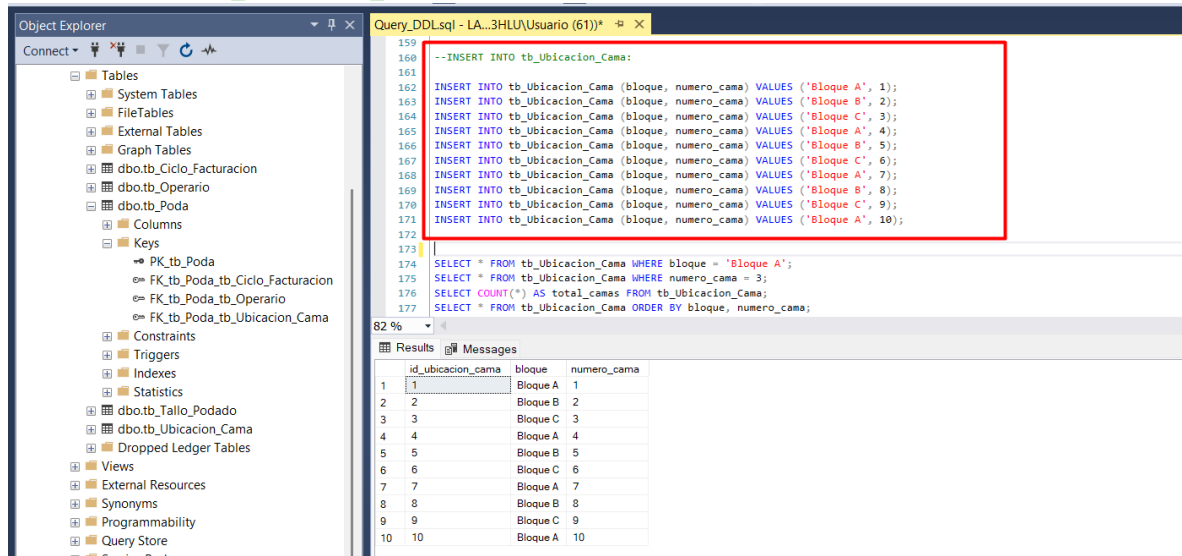
--INSERT INTO tb\_Ciclo\_Facturacion:



```
142
143
144 --INSERT INTO tb_Ciclo_Facturacion:
145
146 INSERT INTO tb_Ciclo_Facturacion (fecha_inicio, fecha_fin) VALUES ('2024-01-01 00:00:00', '2024-03-31 23:59:59');
147 INSERT INTO tb_Ciclo_Facturacion (fecha_inicio, fecha_fin) VALUES ('2024-04-01 00:00:00', '2024-06-30 23:59:59');
148 INSERT INTO tb_Ciclo_Facturacion (fecha_inicio, fecha_fin) VALUES ('2024-07-01 00:00:00', '2024-09-30 23:59:59');
149 INSERT INTO tb_Ciclo_Facturacion (fecha_inicio, fecha_fin) VALUES ('2024-10-01 00:00:00', '2024-12-31 23:59:59');
150 INSERT INTO tb_Ciclo_Facturacion (fecha_inicio, fecha_fin) VALUES ('2025-01-01 00:00:00', '2025-03-31 23:59:59');
151 INSERT INTO tb_Ciclo_Facturacion (fecha_inicio, fecha_fin) VALUES ('2025-04-01 00:00:00', '2025-06-30 23:59:59');
152 INSERT INTO tb_Ciclo_Facturacion (fecha_inicio, fecha_fin) VALUES ('2025-07-01 00:00:00', '2025-09-30 23:59:59');
153 INSERT INTO tb_Ciclo_Facturacion (fecha_inicio, fecha_fin) VALUES ('2025-10-01 00:00:00', '2025-12-31 23:59:59');
154 INSERT INTO tb_Ciclo_Facturacion (fecha_inicio, fecha_fin) VALUES ('2026-01-01 00:00:00', '2026-03-31 23:59:59');
155 INSERT INTO tb_Ciclo_Facturacion (fecha_inicio, fecha_fin) VALUES ('2026-04-01 00:00:00', '2026-06-30 23:59:59');
156
157 SELECT * FROM tb_Ciclo_Facturacion;
158 SELECT COUNT(*) AS total_ciclos_facturacion FROM tb_Ciclo_Facturacion;
159 SELECT * FROM tb_Ciclo_Facturacion WHERE YEAR(fecha_inicio) = 2024;
```

id_ciclo_facturacion	fecha_inicio	fecha_fin
1	2024-01-01	2024-03-31
2	2024-04-01	2024-06-30
3	2024-07-01	2024-09-30
4	2024-10-01	2024-12-31
5	2025-01-01	2025-03-31
6	2025-04-01	2025-06-30
7	2025-07-01	2025-09-30
8	2025-10-01	2025-12-31
9	2026-01-01	2026-03-31
10	2026-04-01	2026-06-30

--INSERT INTO tb\_Ubicacion\_Cama:



Object Explorer

Connect

Tables

- System Tables
- FileTables
- External Tables
- Graph Tables
- dbo.tb\_Ciclo\_Facturacion
- dbo.tb\_Operario
- dbo.tb\_Poda
  - Columns
  - Keys
    - PK\_tb\_Poda
    - FK\_tb\_Poda\_tb\_Ciclo\_Facturacion
    - FK\_tb\_Poda\_tb\_Operario
    - FK\_tb\_Poda\_tb\_Ubicacion\_Cama
- Constraints
- Triggers
- Indexes
- Statistics
- dbo.tb\_Tallo\_Podado
- dbo.tb\_Ubicacion\_Cama
- Dropped Ledger Tables
- Views
- External Resources
- Synonyms
- Programmability
- Query Store

Query\_DDLsql - LA...3HLU\Usuario (61)\*

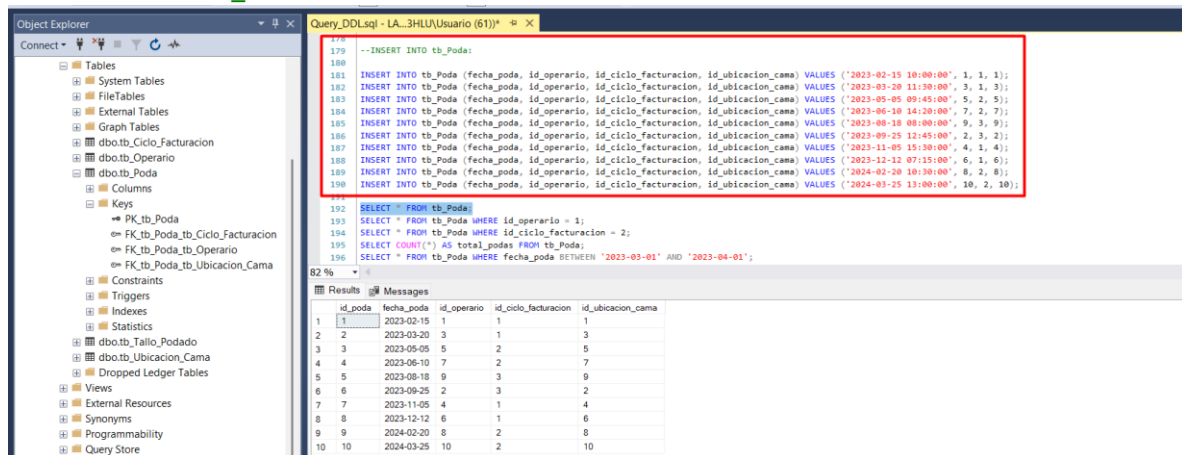
```
--INSERT INTO tb_Ubicacion_Cama:
INSERT INTO tb_Ubicacion_Cama (bloque, numero_cama) VALUES ('Bloque A', 1);
INSERT INTO tb_Ubicacion_Cama (bloque, numero_cama) VALUES ('Bloque B', 2);
INSERT INTO tb_Ubicacion_Cama (bloque, numero_cama) VALUES ('Bloque C', 3);
INSERT INTO tb_Ubicacion_Cama (bloque, numero_cama) VALUES ('Bloque A', 4);
INSERT INTO tb_Ubicacion_Cama (bloque, numero_cama) VALUES ('Bloque B', 5);
INSERT INTO tb_Ubicacion_Cama (bloque, numero_cama) VALUES ('Bloque C', 6);
INSERT INTO tb_Ubicacion_Cama (bloque, numero_cama) VALUES ('Bloque A', 7);
INSERT INTO tb_Ubicacion_Cama (bloque, numero_cama) VALUES ('Bloque B', 8);
INSERT INTO tb_Ubicacion_Cama (bloque, numero_cama) VALUES ('Bloque C', 9);
INSERT INTO tb_Ubicacion_Cama (bloque, numero_cama) VALUES ('Bloque A', 10);

SELECT * FROM tb_Ubicacion_Cama WHERE bloque = 'Bloque A';
SELECT * FROM tb_Ubicacion_Cama WHERE numero_cama = 3;
SELECT COUNT(*) AS total_camas FROM tb_Ubicacion_Cama;
SELECT * FROM tb_Ubicacion_Cama ORDER BY bloque, numero_cama;
```

Results

	id_ubicacion_cama	bloque	numero_cama
1	1	Bloque A	1
2	2	Bloque B	2
3	3	Bloque C	3
4	4	Bloque A	4
5	5	Bloque B	5
6	6	Bloque C	6
7	7	Bloque A	7
8	8	Bloque B	8
9	9	Bloque C	9
10	10	Bloque A	10

--INSERT INTO tb\_Poda:



Object Explorer

Connect

Tables

- System Tables
- FileTables
- External Tables
- Graph Tables
- dbo.tb\_Ciclo\_Facturacion
- dbo.tb\_Operario
- dbo.tb\_Poda
  - Columns
  - Keys
    - PK\_tb\_Poda
    - FK\_tb\_Poda\_tb\_Ciclo\_Facturacion
    - FK\_tb\_Poda\_tb\_Operario
    - FK\_tb\_Poda\_tb\_Ubicacion\_Cama
- Constraints
- Triggers
- Indexes
- Statistics
- dbo.tb\_Tallo\_Podado
- dbo.tb\_Ubicacion\_Cama
- Dropped Ledger Tables
- Views
- External Resources
- Synonyms
- Programmability
- Query Store

Query\_DDLsql - LA...3HLU\Usuario (61)\*

```
--INSERT INTO tb_Poda:
INSERT INTO tb_Poda (fecha_poda, id_operario, id_ciclo_facturacion, id_ubicacion_cama) VALUES ('2023-02-15 10:00:00', 1, 1, 1);
INSERT INTO tb_Poda (fecha_poda, id_operario, id_ciclo_facturacion, id_ubicacion_cama) VALUES ('2023-03-20 11:30:00', 3, 1, 3);
INSERT INTO tb_Poda (fecha_poda, id_operario, id_ciclo_facturacion, id_ubicacion_cama) VALUES ('2023-05-05 09:45:00', 5, 2, 5);
INSERT INTO tb_Poda (fecha_poda, id_operario, id_ciclo_facturacion, id_ubicacion_cama) VALUES ('2023-06-10 14:20:00', 7, 2, 7);
INSERT INTO tb_Poda (fecha_poda, id_operario, id_ciclo_facturacion, id_ubicacion_cama) VALUES ('2023-08-18 08:00:00', 9, 3, 9);
INSERT INTO tb_Poda (fecha_poda, id_operario, id_ciclo_facturacion, id_ubicacion_cama) VALUES ('2023-09-25 12:45:00', 2, 3, 2);
INSERT INTO tb_Poda (fecha_poda, id_operario, id_ciclo_facturacion, id_ubicacion_cama) VALUES ('2023-11-05 15:30:00', 4, 1, 4);
INSERT INTO tb_Poda (fecha_poda, id_operario, id_ciclo_facturacion, id_ubicacion_cama) VALUES ('2023-12-12 07:15:00', 6, 1, 6);
INSERT INTO tb_Poda (fecha_poda, id_operario, id_ciclo_facturacion, id_ubicacion_cama) VALUES ('2024-02-20 10:30:00', 8, 2, 8);
INSERT INTO tb_Poda (fecha_poda, id_operario, id_ciclo_facturacion, id_ubicacion_cama) VALUES ('2024-03-25 13:00:00', 10, 2, 10);

SELECT * FROM tb_Poda;
SELECT * FROM tb_Poda WHERE id_operario = 1;
SELECT * FROM tb_Poda WHERE id_ciclo_facturacion = 2;
SELECT COUNT(*) AS total_poda FROM tb_Poda;
SELECT * FROM tb_Poda WHERE fecha_poda BETWEEN '2023-03-01' AND '2023-04-01';
```

Results

	id_poda	fecha_poda	id_operario	id_ciclo_facturacion	id_ubicacion_cama
1	1	2023-02-15	1	1	1
2	2	2023-03-20	3	1	3
3	3	2023-05-05	5	2	5
4	4	2023-06-10	7	2	7
5	5	2023-08-18	9	3	9
6	6	2023-09-25	2	3	2
7	7	2023-11-05	4	1	4
8	8	2023-12-12	6	1	6
9	9	2024-02-20	8	2	8
10	10	2024-03-25	10	2	10

--INSERT INTO tb\_Tallo\_Podado:

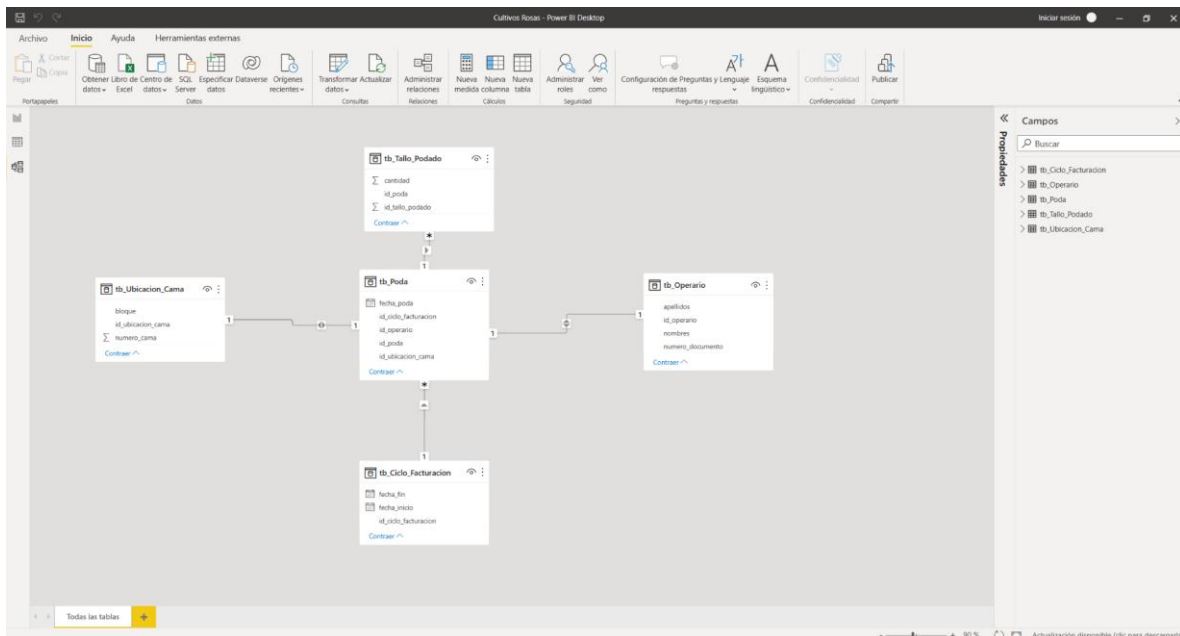
The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure for 'RosaCultivoDB'. The central query window, titled 'Query\_DDLsql - LA...3HLUUsuario (61)\*', contains a series of INSERT statements into the 'tb\_Tallo\_Podado' table. The statements are numbered 201 through 216. A red box highlights these statements. Below the query window, the 'Results' tab shows the data inserted into the table, with columns 'id\_tallo\_podado', 'cantidad', and 'id\_poda'.

id_tallo_podado	cantidad	id_poda
1	20	5
2	15	2
3	30	8
4	25	3
5	18	7
6	22	1
7	12	9
8	28	4
9	17	6
10	23	10
11	18	3
12	26	8
13	14	1
14	21	5
15	19	2

### Punto 3:

Diseñe un reporte de Power BI conectado a la base de datos del punto 2. En dicho reporte se pretende responder las siguientes preguntas:

- Cuántas rosas están sembradas en cada una de las camas.
- Cuántas rosas se podaron la semana pasada por color.
- Cuál es la cama más productiva en el último mes.
- Cuál es el operario más productivo en la poda de los últimos 3 meses.





Tablas Importadas

ArchivoInicioAyudaHerramientas externasHerramientas de tablas

Nombre: tb\_Operario

Marcar como tabla de fechas

Administrar relaciones

Nueva medida

Medida rápida

Nueva columna

Nueva tabla

Calculos

Id\_operario

numero\_documento

nombres

apellidos

1	123456789	Juan	Pérez
2	987654321	Maria	Gómez
3	456789012	Carlos	Rodríguez
4	654321098	Elena	Martínez
5	789012345	Luis	Sánchez
6	210987654	Sophie	Dubois
7	543210987	Pablo	García
8	876543210	Isabella	Moreno
9	109876543	Miguel	López
10	321098765	Olivia	Fernández

Campos

Buscar

tb\_Ciclo\_Facturacion

tb\_Operario

tb\_Poda

tb\_Tallo\_Podado

tb\_Ubicacion\_Cama

Tabla: tb\_Operario (10 filas)

Actualización disponible (clic para descargarla)

Reporte en Power BI:

