

Universidad El Bosque



Facultad de Ingeniería

Programa de Ingeniería de Sistemas

Bases de Datos 1

Proyecto Granja Stardew Valley

Ing. Armando Ricardo Medina Nieto

Jonathan Barrera Fernández

Samuel Andrés Mesa Comas

Juan Felipe Valderrama Peñaloza

Bogotá, Colombia

2025-2

CONTENIDO

Introducción	3
Consideraciones previas.....	3
Consultas SQL	3
Sentencias de álgebra relacional	14
Conclusiones	16

Introducción

El presente documento contiene las sentencias de álgebra relacional diseñadas a partir del modelo relacional implementado para el sistema de información de la Granja Stardew Valley.

Estas expresiones permiten representar de forma matemática las consultas fundamentales que el sistema debe soportar, garantizando formalidad y coherencia con las estructuras de las 16 tablas definidas.

Consideraciones previas

Las tablas utilizadas provienen del modelo relacional final:

- empleado
- maquinaria
- empleado_maquinaria
- animal
- empleado_animal
- cultivo
- empleado_cultivo
- recursos
- consumo_animal
- consumo_cultivo
- producto
- genera_producto_animal
- genera_producto_cultivo
- cliente
- venta
- venta_detalle

Consultas SQL

1. Lista básica de empleados (proyección simple)

```
SELECT identificacion, nombre, cargo, fecha_ingreso FROM empleado;
```

```

1 • SELECT identificacion, nombre, cargo, fecha_ingreso
2 FROM empleado;

```

identificacion	nombre	cargo	fecha_ingreso
E001	Juan Pérez	Veterinario	2021-01-10
E002	Ana Gómez	Agrónomo	2020-03-15
E003	Luis Martínez	Auxiliar	2023-07-01
E004	María López	Administrador	2019-11-20
NULL	NULL	NULL	NULL

2. Empleados con sueldo mayor a 2 millones (selección)
 SELECT identificacion, nombre, cargo, sueldo FROM empleado WHERE sueldo > 2000000;

```

1 • SELECT identificacion, nombre, cargo, sueldo
2 FROM empleado
3 WHERE sueldo > 2000000;

```

identificacion	nombre	cargo	sueldo
E001	Juan Pérez	Veterinario	2500000.00
E002	Ana Gómez	Agrónomo	2300000.00
E004	María López	Administrador	3000000.00
NULL	NULL	NULL	NULL

3. Empleados y la maquinaria que usan (JOIN M:N)
 SELECT e.identificacion, e.nombre, m.id AS id_maquinaria, m.nombre AS nombre_maquinaria, em.asignacion_fecha FROM empleado e JOIN empleado_maquinaria em ON e.identificacion = em.identificacion JOIN maquinaria m ON em.id_maquinaria = m.id;

```

1 • SELECT e.identificacion,
2         e.nombre,
3         m.id AS id_maquinaria,
4         m.nombre AS nombre_maquinaria,
5         em.asignacion_fecha
6 FROM empleado e
7 JOIN empleado_maquinaria em
8     ON e.identificacion = em.identificacion
9 JOIN maquinaria m
10    ON em.id_maquinaria = m.id;

```

	identificacion	nombre	id_maquinaria	nombre_maquinaria	asignacion_fecha
▶	E001	Juan Pérez	1	Tractor John Deere	2024-01-01
	E002	Ana Gómez	2	Fumigadora 1	2024-02-01
	E002	Ana Gómez	3	Arado Principal	2024-03-01
	E003	Luis Martínez	2	Fumigadora 1	2024-04-01

4. Empleados que cuidan vacas vivas (JOIN + selección)

SELECT DISTINCT e.identificacion, e.nombre, a.id AS id_animal, a.nombre AS nombre_animal FROM empleado e JOIN empleado_animal ea ON e.identificacion = ea.identificacion JOIN animal a ON ea.id_animal = a.id WHERE a.especie = 'VACA' AND a.estado = 'VIVO';

```

1 • SELECT DISTINCT e.identificacion,
2         e.nombre,
3         a.id AS id_animal,
4         a.nombre AS nombre_animal
5 FROM empleado e
6 JOIN empleado_animal ea
7     ON e.identificacion = ea.identificacion
8 JOIN animal a
9     ON ea.id_animal = a.id
10 WHERE a.especie = 'VACA'
11 AND a.estado = 'VIVO';

```

	identificacion	nombre	id_animal	nombre_animal
▶	E001	Juan Pérez	1	Vaca 01
	E001	Juan Pérez	2	Vaca 02

5. Empleados que trabajan en cultivos de tipo “HORTALIZA”

SELECT DISTINCT e.identificacion, e.nombre, c.id AS id_cultivo, c.nombre AS nombre_cultivo, c.tipo FROM empleado e JOIN empleado_cultivo ec ON e.identificacion = ec.identificacion JOIN cultivo c ON ec.id_cultivo = c.id WHERE c.tipo = 'HORTALIZA';

```

1 • SELECT DISTINCT e.identificacion,
2     e.nombre,
3     c.id AS id_cultivo,
4     c.nombre AS nombre_cultivo,
5     c.tipo
6 FROM empleado e
7 JOIN empleado_cultivo ec
8     ON e.identificacion = ec.identificacion
9 JOIN cultivo c
10     ON ec.id_cultivo = c.id
11 WHERE c.tipo = 'HORTALIZA';

```

identificacion	nombre	id_cultivo	nombre_cultivo	tipo
E002	Ana Gómez	2	Tomate Invernadero	HORTALIZA
E002	Ana Gómez	3	Lechuga Parcela	HORTALIZA

6. Consumo total de recursos por animal (JOIN + GROUP BY)

SELECT a.id AS id_animal, COALESCE(a.nombre, '(sin nombre)') AS nombre_animal, r.nombre AS recurso, SUM(ca.cantidad) AS total_consumido FROM consumo_animal ca JOIN animal a ON ca.id_animal = a.id JOIN recursos r ON ca.id_recurso = r.id GROUP BY a.id, a.nombre, r.nombre ORDER BY a.id, r.nombre;

```

1 • SELECT a.id AS id_animal,
2     COALESCE(a.nombre, '(sin nombre)') AS nombre_animal,
3     r.nombre AS recurso,
4     SUM(ca.cantidad) AS total_consumido
5 FROM consumo_animal ca
6 JOIN animal a
7     ON ca.id_animal = a.id
8 JOIN recursos r
9     ON ca.id_recurso = r.id
10 GROUP BY a.id, a.nombre, r.nombre
11 ORDER BY a.id, r.nombre;

```

id_animal	nombre_animal	recurso	total_consumido
1	Vaca 01	Agua de Riego	40.00
1	Vaca 01	Concentrado Bovino	110.00
2	Vaca 02	Concentrado Bovino	30.00
4	Cerdo 01	Concentrado Bovino	25.00

7. Consumo total de recursos por cultivo

SELECT c.id AS id_cultivo, COALESCE(c.nombre, '(sin nombre)') AS nombre_cultivo, r.nombre AS recurso, SUM(cc.cantidad) AS total_consumido FROM consumo_cultivo cc

JOIN cultivo c ON cc.id_cultivo = c.id JOIN recursos r ON cc.id_recurso = r.id GROUP BY c.id, c.nombre, r.nombre ORDER BY c.id, r.nombre;

```

1 • SELECT c.id AS id_cultivo,
2         COALESCE(c.nombre, '(sin nombre)') AS nombre_cultivo,
3         r.nombre AS recurso,
4         SUM(cc.cantidad) AS total_consumido
5 FROM consumo_cultivo cc
6 JOIN cultivo c
7     ON cc.id_cultivo = c.id
8 JOIN recursos r
9     ON cc.id_recurso = r.id
10 GROUP BY c.id, c.nombre, r.nombre
11 ORDER BY c.id, r.nombre;

```

Result Grid				
Filter Rows: <input type="text"/>				
Export: <input type="button" value="Export"/> Wrap Cell Content: <input type="button" value="Wrap"/>				
	id_cultivo	nombre_cultivo	recurso	total_consumido
▶	1	Maíz Lote 1	Agua de Riego	100.00
	1	Maíz Lote 1	Fertilizante NPK	10.00
	2	Tomate Invernadero	Agua de Riego	80.00
	3	Lechuga Parcela	Fertilizante NPK	5.00

8. Producción de productos por animales (JOIN)

SELECT a.id AS id_animal, COALESCE(a.nombre, '(sin nombre)') AS nombre_animal, p.identificacion AS id_producto, p.nombre AS producto, SUM(gpa.cantidad) AS total_producido FROM genera_producto_animal gpa JOIN animal a ON gpa.id_animal = a.id JOIN producto p ON gpa.identificacion_producto = p.identificacion GROUP BY a.id, a.nombre, p.identificacion, p.nombre ORDER BY a.id, p.identificacion;

```

1 • SELECT a.id AS id_animal,
2         COALESCE(a.nombre, '(sin nombre)') AS nombre_animal,
3         p.identificacion AS id_producto,
4         p.nombre AS producto,
5         SUM(gpa.cantidad) AS total_producido
6 FROM genera_producto_animal gpa
7 JOIN animal a
8     ON gpa.id_animal = a.id
9 JOIN producto p
10    ON gpa.identificacion_producto = p.identificacion
11 GROUP BY a.id, a.nombre, p.identificacion, p.nombre
12 ORDER BY a.id, p.identificacion;

```

Result Grid					
Filter Rows: <input type="text"/>					
Export: Wrap Cell Content:					
	id_animal	nombre_animal	id_producto	producto	total_producido
▶	1	Vaca 01	P001	Leche Entera	50.00
	2	Vaca 02	P001	Leche Entera	40.00
	5	Gallina 01	P002	Huevos	200.00

9. Producción de productos por cultivos

SELECT c.id AS id_cultivo, COALESCE(c.nombre, '(sin nombre)') AS nombre_cultivo, p.identificacion AS id_producto, p.nombre AS producto, SUM(gpc.cantidad) AS total_producido FROM genera_producto_cultivo gpc JOIN cultivo c ON gpc.id_cultivo = c.id JOIN producto p ON gpc.identificacion_producto = p.identificacion GROUP BY c.id, c.nombre, p.identificacion, p.nombre ORDER BY c.id, p.identificacion;

```

1 • SELECT c.id AS id_cultivo,
2         COALESCE(c.nombre, '(sin nombre)') AS nombre_cultivo,
3         p.identificacion AS id_producto,
4         p.nombre AS producto,
5         SUM(gpc.cantidad) AS total_producido
6 FROM genera_producto_cultivo gpc
7 JOIN cultivo c
8     ON gpc.id_cultivo = c.id
9 JOIN producto p
10    ON gpc.identificacion_producto = p.identificacion
11 GROUP BY c.id, c.nombre, p.identificacion, p.nombre
12 ORDER BY c.id, p.identificacion;

```

Result Grid					
Filter Rows: <input type="text"/>					
Export: Wrap Cell Content:					
	id_cultivo	nombre_cultivo	id_producto	producto	total_producido
▶	1	Maíz Lote 1	P003	Maíz en grano	1000.00
	2	Tomate Invernadero	P004	Tomate fresco	300.00
	3	Lechuga Parcela	P004	Tomate fresco	400.00

10. Producción total por producto (animal + cultivo) usando UNION ALL
 SELECT p.identificacion, p.nombre, SUM(x.cantidad) AS total_producido FROM (SELECT
 identificacion_producto AS id_prod, cantidad FROM genera_producto_animal UNION ALL
 SELECT identificacion_producto AS id_prod, cantidad FROM genera_producto_cultivo) x
 JOIN producto p ON p.identificacion = x.id_prod GROUP BY p.identificacion, p.nombre
 ORDER BY total_producido DESC;

```

1 • SELECT p.identificacion,
2       p.nombre,
3       SUM(x.cantidad) AS total_producido
4 FROM (
5       SELECT identificacion_producto AS id_prod, cantidad
6       FROM genera_producto_animal
7       UNION ALL
8       SELECT identificacion_producto AS id_prod, cantidad
9       FROM genera_producto_cultivo
10    ) x
11 JOIN producto p
12   ON p.identificacion = x.id_prod
13 GROUP BY p.identificacion, p.nombre
14 ORDER BY total_producido DESC;

```

identificacion	nombre	total_producido
P003	Maíz en grano	1000.00
P004	Tomate fresco	700.00
P002	Huevos	200.00
P001	Leche Entera	90.00

11. Detalle de ventas con nombre de cliente y producto
 SELECT v.id AS id_venta, v.fecha, c.nombre AS cliente, p.nombre AS producto, vd.cantidad,
 vd.precio_unit, vd.cantidad * vd.precio_unit AS subtotal FROM venta v JOIN cliente c ON
 v.documento_cliente = c.documento JOIN venta_detalle vd ON v.id = vd.id_venta JOIN
 producto p ON vd.identificacion_producto = p.identificacion ORDER BY v.fecha DESC, v.id;

```

1 • SELECT v.id AS id_venta,
2         v.fecha,
3         c.nombre AS cliente,
4         p.nombre AS producto,
5         vd.cantidad,
6         vd.precio_unit,
7         vd.cantidad * vd.precio_unit AS subtotal
8 FROM venta v
9 JOIN cliente c
10    ON v.documento_cliente = c.documento
11 JOIN venta_detalle vd
12    ON v.id = vd.id_venta
13 JOIN producto p
14    ON vd.identificacion_producto = p.identificacion
15 ORDER BY v.fecha DESC, v.id;

```

	id_venta	fecha	cliente	producto	cantidad	precio_unit	subtotal
▶	2	2025-06-05 15:45:00	Supermercado Central	Tomate fresco	80.00	2000.00	160000.0000
	2	2025-06-05 15:45:00	Supermercado Central	Leche Entera	20.00	2600.00	52000.0000
	1	2025-01-10 10:30:00	Restaurante El Trigal	Leche Entera	30.00	2500.00	75000.0000
	1	2025-01-10 10:30:00	Restaurante El Trigal	Maíz en grano	50.00	1500.00	75000.0000
	3	2024-12-20 09:15:00	Cliente Particular	Huevos	40.00	500.00	20000.0000

12. Total de compras por cliente

SELECT c.documento, c.nombre, SUM(vd.cantidad * vd.precio_unit) AS total_comprado
FROM cliente c JOIN venta v ON c.documento = v.documento_cliente JOIN venta_detalle vd
ON v.id = vd.id_venta GROUP BY c.documento, c.nombre ORDER BY total_comprado
DESC;

```

1 • SELECT c.documento,
2         c.nombre,
3         SUM(vd.cantidad * vd.precio_unit) AS total_comprado
4 FROM cliente c
5 JOIN venta v
6    ON c.documento = v.documento_cliente
7 JOIN venta_detalle vd
8    ON v.id = vd.id_venta
9 GROUP BY c.documento, c.nombre
10 ORDER BY total_comprado DESC;


```

	documento	nombre	total_comprado
▶	C002	Supermercado Central	212000.0000
	C001	Restaurante El Trigal	150000.0000
	C003	Cliente Particular	20000.0000

13. Producto más vendido (en cantidad)

SELECT p.identificacion, p.nombre, SUM(vd.cantidad) AS total_cantidad FROM venta_detalle vd JOIN producto p ON vd.identificacion_producto = p.identificacion GROUP BY p.identificacion, p.nombre ORDER BY total_cantidad DESC LIMIT 1;


```
1 • SELECT p.identificacion,
2       p.nombre,
3       SUM(vd.cantidad) AS total_cantidad
4 FROM venta_detalle vd
5 JOIN producto p
6     ON vd.identificacion_producto = p.identificacion
7 GROUP BY p.identificacion, p.nombre
8 ORDER BY total_cantidad DESC
9 LIMIT 1;
```

Result Grid			
Filter Rows: <input type="text"/>			
Export:  Wrap Cell C			
	identificacion	nombre	total_cantidad
▶	P004	Tomate fresco	80.00

14. Empleados sin maquinaria asignada (LEFT JOIN + selección)

SELECT e.identificacion, e.nombre, e.cargo FROM empleado e LEFT JOIN empleado_maquinaria em ON e.identificacion = em.identificacion WHERE em.identificacion IS NULL;

```
1 • SELECT e.identificacion,
2       e.nombre,
3       e.cargo
4 FROM empleado e
5 LEFT JOIN empleado_maquinaria em
6     ON e.identificacion = em.identificacion
7 WHERE em.identificacion IS NULL;
```

Result Grid			
Filter Rows: <input type="text"/>			
Export: 			
	identificacion	nombre	cargo
▶	E004	María López	Administrador

15. Animales sin registros de producción

SELECT a.id, COALESCE(a.nombre, '(sin nombre)') AS nombre_animal, a.especie, a.estado FROM animal a LEFT JOIN genera_producto_animal gpa ON a.id = gpa.id_animal WHERE gpa.id_animal IS NULL;

```

1 • SELECT a.id,
2         COALESCE(a.nombre, '(sin nombre)') AS nombre_animal,
3         a.especie,
4         a.estado
5 FROM animal a
6 LEFT JOIN genera_producto_animal gpa
7     ON a.id = gpa.id_animal
8 WHERE gpa.id_animal IS NULL;

```

Result Grid				
Filter Rows:				
Export:				
Wrap Cell Content:				
	id	nombre_animal	especie	estado
▶	3	Toro 01	VACA	VIVO
	4	Cerdo 01	CERDO	VIVO
	6	Caballo 01	CABALLO	VIVO

16. Sueldo promedio por cargo

SELECT cargo, AVG(sueldo) AS sueldo_promedio FROM empleado GROUP BY cargo ORDER BY sueldo_promedio DESC;

```

1 • SELECT cargo,
2         AVG(sueldo) AS sueldo_promedio
3 FROM empleado
4 GROUP BY cargo
5 ORDER BY sueldo_promedio DESC;

```

Result Grid		
Filter Rows:		
Export:		
	cargo	sueldo_promedio
▶	Administrador	3000000.000000
	Veterinario	2500000.000000
	Agrónomo	2300000.000000
	Auxiliar	1500000.000000

17. Animales con consumo total de recursos mayor a 100 (HAVING)

SELECT a.id, COALESCE(a.nombre, '(sin nombre)') AS nombre_animal, SUM(ca.cantidad) AS total_consumido FROM animal a JOIN consumo_animal ca ON a.id = ca.id_animal GROUP BY a.id, a.nombre HAVING SUM(ca.cantidad) > 100 ORDER BY total_consumido DESC;

```

1 • SELECT a.id,
2         COALESCE(a.nombre, '(sin nombre)') AS nombre_animal,
3         SUM(ca.cantidad) AS total_consumido
4 FROM animal a
5 JOIN consumo_animal ca
6     ON a.id = ca.id_animal
7 GROUP BY a.id, a.nombre
8 HAVING SUM(ca.cantidad) > 100
9 ORDER BY total_consumido DESC;

```

Result Grid			
Filter Rows: <input type="text"/>			
Export: Wrap Cell Content:			
	id	nombre_animal	total_consumido
▶	1	Vaca 01	150.00

18. Ventas realizadas en un rango de fechas

SELECT v.id, v.fecha, c.nombre AS cliente, v.total FROM venta v JOIN cliente c ON v.documento_cliente = c.documento WHERE v.fecha BETWEEN '2025-01-01' AND '2025-12-31' ORDER BY v.fecha;

```

1 • SELECT v.id,
2         v.fecha,
3         c.nombre AS cliente,
4         v.total
5 FROM venta v
6 JOIN cliente c
7     ON v.documento_cliente = c.documento
8 WHERE v.fecha BETWEEN '2025-01-01' AND '2025-12-31'
9 ORDER BY v.fecha;

```

Result Grid				
Filter Rows: <input type="text"/>				
Export: Wrap Cell Co				
	id	fecha	cliente	total
▶	1	2025-01-10 10:30:00	Restaurante El Trigal	150000.00
	2	2025-06-05 15:45:00	Supermercado Central	212000.00

19. Recursos con stock bajo (alerta de inventario)

SELECT id, nombre, tipo_recurso, stock FROM recursos WHERE stock < 20 ORDER BY stock ASC;

```

1 • SELECT id,
2     nombre,
3     tipo_recurso,
4     stock
5 FROM recursos
6 WHERE stock < 20
7 ORDER BY stock ASC;

```

<				
Result Grid				
Filter Rows:				
	id	nombre	tipo_recurso	stock
▶	4	Vacuna Aftosa	Veterinario	5.00
	3	Fertilizante NPK	Agrícola	15.00
*	NULL	NULL	NULL	NULL

20. Conteo de animales por especie y estado
 SELECT especie, estado, COUNT(*) AS cantidad FROM animal GROUP BY especie, estado
 ORDER BY especie, estado;

```

1 • SELECT especie,
2     estado,
3     COUNT(*) AS cantidad
4 FROM animal
5 GROUP BY especie, estado
6 ORDER BY especie, estado;

```

<			
Result Grid			
Filter Rows:			
	especie	estado	cantidad
▶	CABALLO	VIVO	1
	CERDO	VIVO	1
	GALLINA	VENDIDO	1
	VACA	VIVO	3

Sentencias de álgebra relacional

- Consulta 1: Obtener todos los empleados contratados después del año 2020.
 Propósito: Filtrar empleados por fecha de contratación.

$\sigma_{\text{fecha_ingreso} > '2020-12-31'}(\text{empleado})$

- Consulta 2: Listar las máquinas que están en estado MANTENIMIENTO.
 Propósito: Consultar maquinaria no disponible y pendiente de reparación.

$\sigma_{\text{estado}} = \text{'MANTENIMIENTO'}$ (maquinaria)

- Consulta 3: Obtener todos los animales de especie 'Vaca' que estén vivos.
Propósito: Obtener animales activos de una especie específica.

$\sigma_{\text{especie}} = \text{'VACA'}$ \wedge $\sigma_{\text{estado}} = \text{'VIVO'}$ (animal)

- Consulta 4: Listar los empleados que tienen asignada alguna maquinaria.
Propósito: Identificar empleados con asignación operativa en maquinaria.

$\pi_{\text{empleado.identificacion, nombre, cargo}}$
($\text{empleado} \bowtie_{\text{empleado.identificacion=empleado_maquinaria.identificacion}}$
 $\text{empleado_maquinaria}$)

- Consulta 5: Obtener los cultivos que están listos para cosecha (estado = 'COSECHADO').
Propósito: Saber qué cultivos han completado su ciclo.

$\sigma_{\text{estado}} = \text{'COSECHADO'}$ (cultivo)

- Consulta 6: Listar los recursos con stock menor a 50 unidades.
Propósito: Identificar recursos que requieren reposición.

$\sigma_{\text{stock}} < 50$ (recursos)

- Consulta 7: Obtener el consumo de recursos realizados por animales, mostrando: nombre del animal, recurso consumido y cantidad.
Propósito: Consumos de insumos por animal.

$\pi_{\text{animal.nombre, recursos.nombre, cantidad}}$ (($\text{animal} \bowtie_{\text{animal.id=consumo_animal.id_animal}}$
 consumo_animal) $\bowtie_{\text{consumo_animal.id_recurso=recursos.id recursos}}$)

- Consulta 8: Listar las ventas realizadas y el nombre del cliente asociado.
Propósito: Mostrar las ventas junto a la información del cliente.

$\pi_{\text{venta.id, cliente.nombre, fecha}}$ ($\text{venta} \bowtie_{\text{venta.documento_cliente=cliente.documento cliente}}$)

- Consulta 9: Obtener todos los productos generados por animales (animal \rightarrow producto).
Propósito: Consultar información de producción animal.

$\pi_{\text{animal.nombre, producto.nombre, cantidad}}$
($(\text{animal} \bowtie_{\text{animal.id=genera_producto_animal.id_animal}}$
 $\text{genera_producto_animal}) \bowtie_{\text{genera_producto_animal.identificacion_producto=producto.identi}}$
 ficacion_producto)

- Consulta 10: Calcular la cantidad total vendida por producto.
Propósito: Reporte de ventas por producto.

$\gamma_{\text{identificacion_producto}, \text{SUM}(\text{cantidad}) \rightarrow \text{total_vendido}}(\text{venta_detalle})$

Conclusiones

Las sentencias de álgebra relacional desarrolladas representan operaciones clave que el sistema debe soportar: filtrados, proyecciones, joins, reportes de consumo, producción y ventas. Estas expresiones sirven como base para la implementación en SQL y garantizan consistencia con el modelo relacional del proyecto.